Learning Robust Representations with Score Invariant Learning

Daksh Idnani [1] **Jonathan C. Kao** [2 3]

Abstract

Learning robust representations of data is critical for many machine learning tasks where the test distribution is different from the train distribution. Several recent approaches have proposed new principles to achieve generalizable predictors by learning robust representations from multiple training set distributions. These studies develop new algorithms that guide neural networks to learn representations that are stable across training sets, rather than spurious correlations within any given training set. We develop a new principle for learning robust representations: *score invariant learning* (SIL). We observe that learning a model over multiple train domains can be viewed as learning a *distinct* model distribution over data from each domain, with the property that all model distributions share the same parameters. We use this observation to derive an invariance metric, based on score matching, that learns robust estimators by enforcing invariance of likelihood of points across model distributions. Our experiments demonstrate SIL achieves state-of-the-art performance on the Colored-MNIST task, as well as performance competitive with the state-of-the-art on two domain generalization benchmarks, PACS and VLCS.

1. Introduction

Deep neural networks have demonstrated impressive predictive performance on a variety of machine learning tasks where the train and test data distributions are identical (Krizhevsky et al., 2012; He et al., 2016; Radford et al.; Devlin et al., 2019). However, in many practical applications, the test data distribution differs from the train data distribution (Quionero-Candela et al., 2009). Neural networks trained by minimizing empirical risk over the train distribution often generalize poorly to such test distributions. For example, in image classification, the backgrounds of images may be correlated differently with the classification output in train and test distributions (Beery et al., 2018). In one example, train data may primarily comprise images of birds in the air (blue backgrounds) where as test data may primarily comprise images of birds in trees. A neural network, leveraging this spurious correlation, may learn to classify blue skies as birds and therefore not generalize to examples where birds are not in the sky. In contrast, humans can effortlessly generalize to such shifts in test distribution because they identify features of the object that are invariant across environments, rather than spurious features found in training data (Geirhos et al., 2018). Training neural networks to learn such robust estimators would help many application domains where the test data exhibits different background, noise, or other statistics.

This problem of learning robust representations has lately become an active research area, and several methods have been proposed to improve the ability to generalize to unseen test domains. One approach seeks to incorporate stability properties determined a priori into representations learned by estimators. For example, Carlucci et al. (2019) encode a form of spatial invariance in their classifier by using features extracted by the classifier to solve jigsaw puzzles. These stability properties cause learned representations to achieve better generalization performance across a wide variety of test distributions (Carlucci et al., 2019; Wang et al., 2019a;b). A different approach assumes access to data from multiple training distributions (or "environments") for the same task, and that the difference between train and test distributions can be *extrapolated* from the differences between train distributions. These methods then seek to encode *learned* invariant properties by leveraging the differences in train distributions, through adversarial optimization (Li et al., 2018b; Albuquerque et al., 2019), meta learning (Li et al., 2018a; 2019), or regularized optimization (Arjovsky et al., 2019; Krueger et al., 2020), among other methods (Peters et al., 2016).

Out of these approaches, we choose to focus on learning invariant representations from multiple domains through regularized optimization. This approach benefits from being simple to apply to any risk-minimization problem with data from multiple domains, without posing any new architectural constraints or challenging optimization problems,

[1]Dept of Computer Science, University of California, Los Angeles [2]Dept of Electrical and Computer Engineering, University of California, Los Angeles [3]Neurosciences Program, University of California, Los Angeles. Correspondence to: Daksh Idnani <dakshidnani@ucla.edu>.

such as with meta-learning or adversarial optimization. Further, there are no additional hyperparameters involved in this approach, save for a regularization strength coefficient. This is important because tuning hyperparameters in the absence of the actual test distribution is not a trivial problem. Recently, Arjovsky et al. (2019) proposed an important method in this class of solutions, to learn invariant predictors between domains by using an efficient regularization penalty that encourages learned representations to be simultaneously optimal for all domains. Improving upon this work, Krueger et al. (2020) proposed risk-extrapolation, a method that elicits predictors that attain similar risk across all training domains. These works follow the intuition that predictors should be similarly suboptimal between domains, since the optimality of a stable predictor depends only on non-spurious features present in all domains. It is an open question how to effectively capture this intuition in a single regularization term. In this work, we find a regularization penalty that achieves stronger generalization performance than previous methods of this type, and is robust to some failure cases that exist in previous works, as discussed in the following section. Specifically, we observe that learning a prediction rule for a task over multiple distributions induces unnormalized model distributions over each domain. Working with the assumption that the density of each point should depend only on stable features, and not any domain-specific features, we derive a differentiable and invariant property over the model distributions to match across domains.

2. Background

2.1. Multi-environment invariant representation learning

We consider the setting where we have access to samples from multiple training distributions $(x_i^e, y_i^e) \sim P^e(X, Y)$ where $e \in \{e_1, \cdots, e_n\}$, with $x_i^e \in \mathcal{X}^e$, $y_i^e \in \mathcal{Y}$, and $\mathcal{X}^e \subset \mathcal{X}$. We refer to each distribution as a *domain* or *environment*. We also have some parametric prediction rule $f_\theta : \mathcal{X} \to \mathcal{Y}$, as well as a loss function $\ell(f_\theta(x), y)$, and must learn a single set of parameters θ that minimizes the loss over all training distributions.

First, consider the empirical risk minimization (ERM) framework, in which one minimizes the empirical risk over samples from all environments. The empirical risk over N samples from a single environment is $\mathcal{R}^e(\theta) = \frac{1}{N} \sum_{(x_i^e, y_i^e)} \ell(f_\theta(x_i^e), y_i^e)$. ERM therefore minimizes $\sum_e \mathcal{R}^e(\theta)$. ERM does not make any use of domain information, and behaves as though all samples were drawn from a single training environment. Depending on the make up of the environments, this can lead to the formation of unstable representations, since ERM does not use information about what signals are spurious across environments.

To address this issue, Arjovsky et al. (2019) propose Invariant Risk Minimization (IRM), an approach that learns *invariant representations*, $\Phi(X)$, by considering spurious information across environments. IRM learns $\Phi(X)$ by assuming that, for any stable predictor of the from $w \cdot \Phi : \mathcal{X} \to \mathcal{Y}$ where $w : \mathcal{H} \to \mathcal{Y}$ is a linear transformation, w is simultaneously optimal for all environments. This results in the following bi-leveled optimization problem:

$$\min_{\substack{\Phi:\mathcal{X}\to\mathcal{H} \\ w:\mathcal{H}\to\mathcal{Y}}} \sum_e \mathcal{R}^e(w \cdot \Phi), \text{ subject to} \tag{1}$$

$$w \in \arg\min_{w^*:\mathcal{H}\to\mathcal{Y}} \mathcal{R}^e(w^* \cdot \Phi), \text{ for all } e \in \{e^1, \cdots, e^n\}. \tag{2}$$

Since solving this constrained optimization problem is non-trivial for many problems, Arjovsky et al. (2019) propose to phrase the task as that of minimizing the risks over all environment along with an efficient penalty function added on as a regularization term, $\mathcal{R}_{\text{IRM}} = \sum_e \mathcal{R}^e(\Phi) + \lambda \cdot \mathbb{D}(w, \Phi)$, where $\mathbb{D}(w, \Phi)$ is the regularization term and λ is a hyperparameter. They ultimately find that a good choice for the penalty is $\mathbb{D}(w, \Phi) = \sum_e \|\nabla_{w|w=1.0} \mathcal{R}^e(w \cdot \Phi)\|$.

Following the regularized optimization framework used in IRM, Krueger et al. (2020) proposed a different penalty, using the assumption that training risks across environments should be similar across environments. Krueger et al. (2020) called this approach risk extrapolation (REx). They implemented this assumption by incorporating a regularization term that minimizes the variance of the expected loss across environments (V-REx): $\mathcal{R}_{\text{V-REx}} = \sum_e \mathcal{R}^e + \lambda \cdot \text{Var}(\{\mathcal{R}^{e_1}, \cdots, \mathcal{R}^{e_n}\})$. In cases where each environment has a similar base risk level, this approach regularizes so that the prediction rule is similarly suboptimal across environments. V-REx would aim therefore to not learn spurious correlations that would impact relative performance between environments. In general, noise in the signal may differ across environments, translating into different baseline risk across environments. Krueger et al. (2020) therefore proposed an alternate method where a baseline risk is approximated for each environment using methods proposed by Meinshausen & Bühlmann (2015), with the variance computed over the difference of an environment's risk and its baseline. A limitation of this approach is that this approximation provides only an upper bound on the true noise level, and as such may overestimate the baseline risk.

2.2. Unnormalized statistical models and score matching

In this work, we propose score invariant learning, based on score matching in unnormalized statistical models. In score matching, we consider a setting where $x \in \mathcal{X}$ is an input signal sampled from a data distribution $p(x)$. For some scalar energy function $E_\theta : \mathcal{X} \to \mathbb{R}$, we obtain an energy-based model $p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}$ where $Z(\theta) = \int_{\mathcal{X}} \exp(-E_\theta(x))dx$ is the partition function (LeCun et al., 2006). The log likelihood of an energy-based

model is $\log p_\theta(x) = -E_\theta(x) - \log Z(\theta)$. Maximum-likelihood learning for an energy-based model would therefore decrease the energy of samples $E_\theta(x)$ from the distribution, and increase the total energy over all points, $Z(\theta)$. Maximizing this term requires computing $Z(\theta)$ which is analytically intractable. Many learning algorithms have thus suggested approximating $Z(\theta)$ (Hinton, 2002; Hinton & Salakhutdinov, 2006; Tieleman, 2008) or avoiding computing $Z(\theta)$ (Hyvärinen, 2005; Gutmann & Hyvärinen, 2010; Gutmann & Hyvärinen, 2013). One popular method, score matching (Hyvärinen, 2005), matches the score function of a point in the model distribution with an approximation of the score of the point in the data distribution, using the insight that $\nabla_x Z(\theta) = 0$. This leads to the score of a point, defined as $s_\theta(x) = \nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x)$. By matching the gradients of the log-likelihood, score matching enables learning an unnormalized parametric distribution by accurately representing the *shape* of the data distribution, without needing to compute the irrelevant scaling factor captured by the partition function.

3. Methods

3.1. Probabilistic interpretation of multi-domain learning

We begin by observing that learning a set of parameters θ to minimize the risks $\mathcal{R}^e$ over multiple environments can be interpreted as learning unnormalized statistical models for each environment. This entails maximizing the likelihood of samples from the train data distributions in each environment's model distribution. For each environment, we define an energy-based model with the energy function $E_\theta(X, Y) = \ell(f_\theta(X), Y)$, so that the model density function for an environment is expressed as $P^e_\theta(X, Y) = \frac{\exp(-\ell(f_\theta(X),Y))}{Z^e(\theta)}$, where $Z^e(\theta) = \int_{\mathcal{H}^e} \exp(-\ell(f_\theta(X), Y))\, dX dY$. In ERM over a single-domain, learning θ by minimizing traditional loss functions, including mean-square error and cross-entropy, implicitly increases the partition function. For instance, in a regression task, changing the parameters to decrease the mean-square error at point (x, y) will increase the error (energy) at points (x, y'), where $y' \neq y$. As such, the partition function is typically not a consideration in ERM. In contrast, in our multi-domain learning setting, each environment's partition function is parametrized by the same θ, but take on different values because they integrate over different data domains, $\mathcal{H}^e = (\mathcal{X}^e, \mathcal{Y}^e)$. Thus, decreasing the energy in environment e at a single point, $E_\theta(X, Y)$, may unpredictably impact the partition function in environment e', $Z^{e'}(\theta)$. Further, each partition function captures domain-specific information such as the baseline loss (energy) level over a domain. A robust risk minimization algorithm in a multi-domain setting must account for differences in partition functions between environments in order to make any comparison of model likelihood.

3.2. Score Invariant Learning (SIL)

We propose a differentiable metric, $I_\theta(e)$, capturing the desired invariance that data points in different environments should on average be equally likely. The naïve approach of setting $I_\theta(e) = \mathbb{E}_{(x^e,y^e)\sim P^e}[P^e_\theta(x^e, y^e)]$ to match the average density of points across model density distributions is intractable, since it requires knowing the partition function $Z^e(\theta)$ for each domain. Instead, we avoid needing to approximate the partition function by basing our invariance property on score matching: we consider the gradients of the log densities of the model distributions, thereby considering the *shapes* of the different unnormalized model distributions. By using an invariance property that does not depend on $Z^e(\theta)$, we also avoid needing to factor for the baseline loss level of an environment, a problem in several previous approaches (Ben-Tal et al., 2009; Meinshausen & Bühlmann, 2015; Krueger et al., 2020). To arrive at our invariance property, we first compute the score in each environment as $s^e_\theta(x^e, y^e) = -\nabla_{x^e} E_\theta(x^e, y^e)$. In general, each environment can have different supports, so that we have no correspondence between points from different domains. We must therefore determine how to pool together the score information for each point into a single metric that can be compared between domains. We propose to take the norm of the concatenated score tensor over all samples in a domain, $\|s^e_\theta(x^e, y^e)\|$, which quantifies *how much* small changes in the inputs affect the risk for each environment. This addresses the issue of potentially misaligned supports (or misaligned sensitive regions in the inputs) between environments. To encourage invariance of this property between domains, we minimize the standard deviation of this property individually computed over each domain, resulting in

Table 1. Performance on Colored MNIST with 25% label flipping

Algorithm	Train acc.	Test acc.
SIL (Ours)	69.6 ± 1.3	$\mathbf{70.4 \pm 1.8}$
V-REx	71.5 ± 1.0	68.7 ± 0.9
IRM	70.8 ± 0.9	66.9 ± 2.5
ERM	87.4 ± 0.2	17.1 ± 0.6
Hypothetical optimum	75	75
Grayscale model (Oracle)	73.5 ± 0.2	73.0 ± 0.4

Table 2. Performance on Colored MNIST without label flipping

Algorithm	Train acc.	Test acc.
SIL (Ours)	96.0 ± 0.3	$\mathbf{96.5 \pm 0.3}$
V-REx	97.6 ± 0.3	95.5 ± 0.8
ERM	99.3 ± 0.1	92.5 ± 0.6
IRM	97.1 ± 0.1	83.6 ± 0.8
Hypothetical optimum	100	100
Grayscale model (Oracle)	98.7 ± 0.1	97.7 ± 0.1

Table 3. Performance on VLCS

Algorithm	Caltech	SUN	Pascal	LabelMe	Average
SIL (Ours)	96.89 ± 0.31	65.52 ± 0.83	72.11 ± 0.63	61.66 ± 0.59	74.04
V-REx	96.72	63.68	72.41	60.40	73.30
IRM	95.99	62.85	71.71	59.61	72.54
ERM	94.76	61.92	69.03	60.55	71.56
Adv. target-invariance	95.92	69.37	71.14	67.63	75.92
Jigsaw	96.93	64.30	70.62	60.19	73.19

Table 4. Performance on PACS

Algorithm	Art Painting	Cartoon	Sketch	Photo	Average
SIL (Ours)	69.62 ± 0.47	71.41 ± 1.35	63.10 ± 1.19	89.96 ± 0.47	73.52
V-REx	67.04	67.97	59.81	89.74	71.14
IRM	67.05	68.49	57.81	89.39	70.69
ERM	66.22	67.59	57.90	89.64	70.35
Adv. target-invariance	66.60	73.36	68.03	88.12	74.02
Jigsaw	67.63	71.71	65.18	89.00	73.38

similarly steep model distributions over each domain. Thus, given the sets of samples $(x^e, y^e) \sim P^e(x, y)$, and invariance penalty $I_\theta(e) = \|s^e_\theta(x^e, y^e)\|$, we have:

$$\mathcal{R}_{\text{SIL}}(\theta) = \sum_e \mathcal{R}^e + \lambda \cdot \text{Var}(\{I_\theta(e_1), \cdots, I_\theta(e_n)\})^{\frac{1}{2}}. \quad (3)$$

We note a second intuition for this invariance penalty. Following Heinze-Deml & Meinshausen (2017), we can disentangle the latent features of input X into core features X^{core} that don't change across domains, and extraneous features X^{style} that can be spuriously correlated with the output across domains. While the gradients of the output (and thus the loss) with respect to core features X^{core} remain identical between domains, the gradients with respect to the extraneous features may change dramatically between domains. A predictor that extracts only core features would achieve the same score across all environments, and would therefore minimize our invariance penalty.

4. Experiments

4.1. Colored MNIST experiments

Colored MNIST is a multi-environment binary classification task proposed by Arjovsky et al. (2019) based on the MNIST dataset, where digits are colored red or green, and the correlation between color and output is spurious between environments. The two classes are digits 0-4 or digits 5-9, and there are two train environments along with a single test environment. Arjovsky et al. (2019) further design this experiment so that the train environments are strongly (but spuriously) correlated with color signal, and the test environment is inversely correlated with the same color signal. Specifically, the two train environments are colored such that green digits have probability 0.8 and 0.9 of being digits 5-9, while the test environment is colored such that green digits only have probability 0.1 of being digits 5-9. Additionally, Arjovsky et al. (2019) flip 25% of the class labels before assigning color, resulting in the color signal being a stronger (but spurious) indicator of the label for the train environments. Thus, this task tests if algorithms are able to disregard spurious correlations at training time even if they are better indicators of the output than the stable signal (digit shape). We also evaluated all methods on the same task without label flipping, to study the setting where the stable signal is stronger than spurious indicators, but a substantial amount of spurious correlations are present in the data. We present the mean and standard deviation of the train and test accuracies over ten runs, demonstrating that SIL outperforms IRM and V-REx on Colored MNIST.

4.2. PACS and VLCS

We also evaluated our method on two multi-domain image classification tasks. PACS (Li et al., 2017) and VLCS (Torralba & Efros, 2011; Khosla et al., 2012) are 4-domain classification tasks with 7 and 5 classes respectively. Each dataset presents four domain generalization tasks, training on three domains and testing on the fourth domain. We compare our methods to ERM, IRM, and REx, and provide for reference the performance of the previous and current state-of-the-art methods that use explicitly encoded spatial invariance (jigsaw) (Carlucci et al., 2019) and adversarial optimization (adversarial target-invariance) (Albuquerque et al., 2019) respectively. We use the same architecture, data augmentation, and learning schedule as Carlucci et al. (2019) and Krueger et al. (2020) for sound comparison. Albuquerque et al. (2019) use additional data augmentation and a longer training schedule, so a direct comparison to this method is not applicable. We present the mean and standard deviation of the test accuracy over five runs, and for each run consider the test accuracy obtained on the epoch with highest validation accuracy. On average, SIL outperformed ERM, IRM, and V-REx on both VLCS and PACS.

References

Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T. H., and Mitliagkas, I. Adversarial target-invariant representation learning for domain generalization, 2019.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 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.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7538–7550. Curran Associates, Inc., 2018.

Gutmann, M. and Hyvärinen, A. Estimation of unnormalized statistical models without numerical integration. 2013.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/gutmann10a.html.

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

Heinze-Deml, C. and Meinshausen, N. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14 (8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://doi.org/10.1162/089976602760128018.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075. doi: 10.1126/science.1127647. URL https://science.sciencemag.org/content/313/5786/504.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, December 2005. ISSN 1532-4435.

Khosla, A., Zhou, T., Malisiewicz, T., Efros, A., and Torralba, A. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, Florence, Italy, October 2012.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex), 2020.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. In Bakir, G., Hofman, T., Schölkopf, B., Smola, A., and Taskar, B. (eds.), *Predicting Structured Data*. MIT Press, 2006.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018a.

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1446–1455, 2019.

Li, H., Jialin Pan, S., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b.

Meinshausen, N. and Bühlmann, P. Maximin effects in inhomogeneous large-scale data. *Ann. Statist.*, 43(4): 1801–1830, 08 2015. doi: 10.1214/15-AOS1325. URL https://doi.org/10.1214/15-AOS1325.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1064–1071, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390290. URL https://doi.org/10.1145/1390156.1390290.

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

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 10506–10518. Curran Associates, Inc., 2019a.

Wang, H., He, Z., Lipton, Z. L., and Xing, E. P. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=rJEjjoR9K7.

Appendix

Experiment implementation details

In this paper, we evaluated our method on three datasets: Colored MNIST, PACS, and VLCS. Here, we present architecture and hyperparameter details for all experiments.

For both Colored MNIST experiments, we selected our architecture and hyperparameter search space based on those of Arjovsky et al. (2019). In particular, the neural network architecture is a fully connected net with layer sizes $D \rightarrow H \rightarrow H \rightarrow 1$ where D is the input dimensionality and H is a hyperparameter. We trained both tasks using a negative log likelihood loss and batch gradient descent. The 25% label flipping experiment used 1500 training iterations, while the non-label flipping experiment used 500 iterations. We tuned the learning rate, hidden layer size, the penalty coefficient, and the number of warmup ERM iterations over 25 runs with randomly selected configurations. The learning rate was set to 10^x where x is drawn from a uniform distribution over $[-2.5, -3.5]$. The hidden layer size was drawn from a uniform distribution of integers between 256 and 512. The penalty weight was a random integer between 1 and 10^5, and the number of warmup ERM iterations was a random integer between 100 and 200. Our final choice of hyperparameters for the label flipping experiment was a learning rate of $7.35e-4$, hidden layer size of 321, penalty coefficient of 23921, and 171 warmup iterations. On the experiment without label flipping, we used a learning rate of $6.93e-4$, hidden layer size of 461, penalty coefficient of 44431, and 162 warmup iterations. We also found that this method was not very sensitive to hyperparameters, as many of our runs presented similar final results.

For PACS and VLCS experiments, we used the same network architecture as Carlucci et al. (2019), and additionally used the same data augmentation, optimizer, and learning rate schedule, thus needing to tune only batch size, learning rate, and SIL penalty coefficient. We used a pretrained AlexNet network with the last layer removed as a feature extractor, followed by a single fully connected layer. We trained using SGD for 30 epochs, with the learning rate reduced by a factor of 0.1 for the final 6 epochs. For PACS, we tuned all three hyperparameters by considering test accuracy on the art-painting domain over 25 randomly selected configurations: learning rate was selected out of $\{0.0001, 0.0005, 0.001\}$, batch size was selected out of $\{64, 128, 256\}$, and the penalty coefficient was selected out of $\{0.25, 0.5, 0.75, \ldots, 2.5\}$. For VLCS, we kept the batch size and learning rate found in our PACS tuning, and only tuned the penalty coefficient (selected from the same set) over five runs on the LabelMe domain. Our final choice of hyperparameters were a batch size of 128, a learning rate of 0.0005, and penalty coefficients of 1.5 and 0.5 respectively on PACS and VLCS.