Acknowledgements

The authors want to thank Leon Bottou and Ishaan Gulrajani for their help with IRM. In addition we want to thank David Lopez-Paz, Jörn-Henrik Jacobsen, Ben Glocker, Nick Pawlowski, and Daniel Castro for useful discussions.

Maximilian Ilse was funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Grant DLMedIa: Deep Learning for Medical Image Analysis).

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant Risk Minimization. *arXiv*, 2019.

Azulay, A. and Weiss, Y. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv*, 2019.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. MetaReg: Towards Domain Generalization using Meta-Regularization. In *NeurIPS*. 2018.

Bareinboim, E. and Pearl, J. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, July 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1510507113. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1510507113.

Carlucci, F. M., Russo, P., Tommasi, T., and Caputo, B. Hallucinating Agnostic Images to Generalize Across Domains. *arXiv*, 2018.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain Generalization by Solving Jigsaw Puzzles. *arXiv*, 2019.

Castro, D. C., Walker, I., and Glocker, B. Causality matters in medical imaging. *arXiv*, 2019.

Cohen, T. S. and Welling, M. Group Equivariant Convolutional Networks. *arXiv:1602.07576 [cs, stat]*, June 2016. URL http://arxiv.org/abs/1602.07576. arXiv: 1602.07576.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian Neural Networks. *arXiv:2003.04630 [physics, stat]*, March 2020. URL http://arxiv.org/abs/2003.04630. arXiv: 2003.04630.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. AutoAugment: Learning Augmentation Strategies From Data. In *CVPR*, 2019.

Ding, Z. and Fu, Y. Deep Domain Generalization With Structured Low-Rank Constraint. *IEEE Transactions on Image Processing*, 27(1):304–313, 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-Adversarial Training of Neural Networks. *arXiv*, 2016.

Ghifary, M., Kleijn, W. B., Zhang, M., and Balduzzi, D. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *ICCV*, 2015.

Gontijo-Lopes, R., Smullin, S. J., Cubuk, E. D., and Dyer, E. Affinity and Diversity: Quantifying Mechanisms of Data Augmentation. *arXiv:2002.08973 [cs, stat]*, February 2020. URL http://arxiv.org/abs/2002.08973. arXiv: 2002.08973.

Gowal, S., Qin, C., Huang, P.-S., Cemgil, T., Dvijotham, K., Mann, T., and Kohli, P. Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations. *arXiv*, 2019.

Heinze-Deml, C. and Meinshausen, N. Conditional Variance Penalties and Domain Shift Robustness. *arXiv*, 2019.

Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. DIVA: Domain Invariant Variational Autoencoders. *arXiv*, 2019.

Johansson, F. D., Sontag, D., and Ranganath, R. Support and Invertibility in Domain-Invariant Representations. *arXiv*, 2019.

Khosla, A., Zhou, T., Malisiewicz, T., Efros, A. A., and Torralba, A. Undoing the Damage of Dataset Bias. In *ECCV*. Berlin, Heidelberg, 2012.

Kipf, T. N. and Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, February 2017. URL http://arxiv.org/abs/1609.02907. arXiv: 1609.02907.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*. 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). *arXiv:2003.00688 [cs, stat]*, March 2020. URL http://arxiv.org/abs/2003.00688. arXiv: 2003.00688.

LeCun, Y., Bottou, L., Bengio, Y., and Ha, P. Gradient-Based Learning Applied to Document Recognition. 1998.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, Broader and Artier Domain Generalization. In *ICCV*, October 2017a.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Learning to Generalize: Meta-Learning for Domain Generalization. *arXiv*, 2017b.

Li, S. Y. Automating Data Augmentation: Practice, Theory and New Direction, April 2020. URL http://ai.stanford.edu/blog/data-augmentation/. Library Catalog: ai.stanford.edu.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *ECCV*. 2018.

Lyle, C., van der Wilk, M., Kwiatkowska, M., Gal, Y., and Bloem-Reddy, B. On the Benefits of Invariance in Neural Networks. *arXiv:2005.00178 [cs, stat]*, April 2020. URL http://arxiv.org/abs/2005.00178. arXiv: 2005.00178.

Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10846–10856. Curran Associates, Inc., 2018.

Mancini, M., Bulò, S. R., Caputo, B., and Ricci, E. Best sources forward: domain generalization through source-specific nets. *arXiv*, 2018.

Mooij, J. M., Magliacane, S., and Claassen, T. Joint Causal Inference from Multiple Contexts. *arXiv:1611.10351 [cs, stat]*, April 2019. URL http://arxiv.org/abs/1611.10351. arXiv: 1611.10351.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified Deep Supervised Domain Adaptation and Generalization. *arXiv*, 2017.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain Generalization via Invariant Feature Representation. *arXiv:1301.2115 [cs, stat]*, January 2013. URL http://arxiv.org/abs/1301.2115.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv: 1912.01703.

Pearl, J. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

Perez, L. and Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv*, 2017.

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.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

Rojas-Carulla, M., Scholkopf, B., Turner, R., and Peters, J. Invariant Models for Causal Transfer Learning. pp. 34, 2018.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., and Sarawagi, S. Generalizing Across Domains via Cross-Gradient Training. *arXiv*, 2018.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Subbaswamy, A. and Saria, S. From development to deployment: dataset shift, causality, and shift-stable models in health AI. *Biostatistics*, pp. kxz041, November 2019. ISSN 1465-4644, 1468-4357. doi: 10.1093/biostatistics/kxz041. URL https://academic.oup.com/biostatistics/advance-article/doi/10.1093/biostatistics/kxz041/5631850.

Subbaswamy, A., Schulam, P., and Saria, S. Preventing Failures Due to Dataset Shift: Learning Predictive Models That Transport. *arXiv:1812.04597 [cs, stat]*, February 2019. URL http://arxiv.org/abs/1812.04597. arXiv: 1812.04597.

Tellez, D., Litjens, G., Bandi, P., Bulten, W., Bokhorst, J.-M., Ciompi, F., and van der Laak, J. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *arXiv*, 2019.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *arXiv*, 2017.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv*, 2014.

Vapnik, V. Principles of Risk Minimization for Learning Theory. In *NIPS*. 1992.

Wang, H., He, Z., Lipton, Z. C., and Xing, E. P. Learning Robust Representations by Projecting Superficial Statistics Out. In *ICLR*, 2018.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv*, 2018.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On Learning Invariant Representation for Domain Adaptation. *arXiv*, 2019.

A. Appendix

A.1. Additional details for SDA

All data augmentations are implemented using the TORCHVISION.TRANSFORMS module of PyTorch (Paszke et al., 2019). We choose the range of the hyperparameters of the augmentations in such a way that they do not destroy all information in x, e.g., setting the brightness of all pixels to 0 or translating all pixels by the full image width. In all experiments we use the following data augmentations:

- 'brightness':
  ```
  torchvision.transforms.ColorJitter(brightness=1.0,
  contrast=0, saturation=0, hue=0)
  ```
- 'contrast':
  ```
  torchvision.transforms.ColorJitter(brightness=0,
  contrast=10.0, saturation=0, hue=0)
  ```
- 'saturation':
  ```
  torchvision.transforms.ColorJitter(brightness=0, contrast=0,
  saturation=10.0, hue=0)
  ```
- 'hue':
  ```
  torchvision.transforms.ColorJitter(brightness=0, contrast=0,
  saturation=0, hue=0.5)
  ```
- 'rotation':
  ```
  torchvision.transforms.RandomAffine([0, 359],
  translate=None, scale=None, shear=None,
  resample=PIL.Image.BILINEAR, fillcolor=0)
  ```
- 'translate':
  ```
  torchvision.transforms.RandomAffine(0, translate=[0.2,
  0.2], scale=None, shear=None, resample=PIL.Image.BILINEAR,
  fillcolor=0)
  ```
- 'scale':
  ```
  torchvision.transforms.RandomAffine(0, translate=None,
  scale=[0.8, 1.2], shear=None, resample=PIL.Image.BILINEAR,
  fillcolor=0)
  ```
- 'shear':
  ```
  torchvision.transforms.RandomAffine(0, translate=None,
  scale=None, shear=[-10., 10., -10., 10.],
  resample=PIL.Image.BILINEAR, fillcolor=0)
  ```
- 'vflip':
  ```
  torchvision.transforms.RandomVerticalFlip(p=0.5)
  ```
- 'hflip':
  ```
  torchvision.transforms.RandomHorizontalFlip(p=0.5)
  ```

A.1.1. Ablation study on rotated MNIST

We will demonstrate now that SDA can also be used to find the most suitable hyperparameters for the data augmentations used in this paper. In this example we focus on the rotated MNIST dataset and the data augmentation 'rotate'. We use the same experimental setup as described in the rotated MNIST experiment. We choose $\{30°, 60°, 90°\}$ as the training domains and $0°$ as the test domain. We compare five sets of hyperparameters, where each set defines the range from which the rotation angle is uniformly sampled. In Table 4, we find that the hyperparameters $[0°, 359°]$ lead to the lowest domain accuracy, i.e., simulate an intervention on h_d the best.

Table 4. Comparing domain accuracy on rotated MNIST for five different sets of the data augmentation 'rotate'. Average ± standard error over five seeds.

Hyperparameter	domain accuracy
$[-15°, 15°]$	92.60 ± 0.98
$[-45°, 45°]$	82.63 ± 0.89
$[-90°, 90°]$	69.79 ± 0.91
$[0°, 180°]$	63.16 ± 1.51
$[0°, 359°]$	51.70 ± 2.21

A.1.2. Results of domain classifier on each dataset

For each dataset, we train a domain classifier using the same architecture and training procedure as used for the label classifier. We only use samples from the training domains and repeat each experiment five times. In Table 5, we show the domain accuracy for each of the datasets. In the case of rotated MNIST, we perform four experiments where each of the domains $d = \{0°, 30°, 60°, 90°\}$ is used for testing once, while the remaining three domains are used for training. For each individual experiment SDA returns the augmentation 'rotate' as the most suitable. In Table 5, we show the average of the four experiments that where each repeated five times. In the case of colored MNIST, the training and test domains are fixed therefore we only conducted one experiment. We show the average of the one experiment that was repeated five times. For PACS, we perform four experiments where each of the domains $d = \{\text{'photo', 'art painting', 'cartoon', 'sketch'}\}$ is used for testing once, while the remaining three domains are used for training. We use cross validation over all four experiments to select the data augmentation. In Table 5, we show the average of the four experiments that where each repeated five times.

A.2. Colored MNIST

The DAG of the data generating process for the colored MNIST experiment is shown in Figure 6 (left), where d is

Table 5. Domain accuracy for each dataset. Average ± standard error.

Data Augmentation	rotated MNIST	Colored MNIST	PACS
'brightness'	98.45 ± 0.24	50.1524 ± 0.1527	96.46 ± 0.37
'contrast'	98.64 ± 0.23	50.1470 ± 0.0506	96.41 ± 0.37
'saturation'	98.95 ± 0.21	50.1894 ± 0.0593	96.03 ± 0.43
'hue'	98.66 ± 0.36	50.0006 ± 0.0028	96.32 ± 0.41
'rotation'	64.70 ± 2.21	50.0024 ± 0.0030	96.59 ± 0.39
'translation'	90.84 ± 1.65	50.0004 ± 0.0008	96.82 ± 0.34
'scale'	91.42 ± 1.34	50.2082 ± 0.1327	97.00 ± 0.29
'shear'	91.48 ± 1.14	50.2252 ± 0.1531	96.82 ± 0.34
'vertical flip'	88.79 ± 0.50	50.1560 ± 0.0140	96.88 ± 0.34
'horizontal flip'	91.98 ± 0.29	50.4060 ± 0.0274	96.54 ± 0.33

the domain, y is the binary label, $\hat{y}$ is the original MNIST label, h_d are high-level color features caused by d and y, h_y are high-level shape features caused by $\hat{y}$, and x is the observed image. In the case of the colored MNIST dataset the spurious correlation between d and y is the result of the collider h_d (that itself is a parent of the observed node x). While the cause of the spurious correlation between d and y is different, the reasoning in Section 2 is still valid. In Figure 6 (right), we show that in theory an intervention on h_d will remove the spurious correlation between d and y. We argue that an intervention on h_d can be simulated by data augmentation, we present experimental evidence in Section 4.

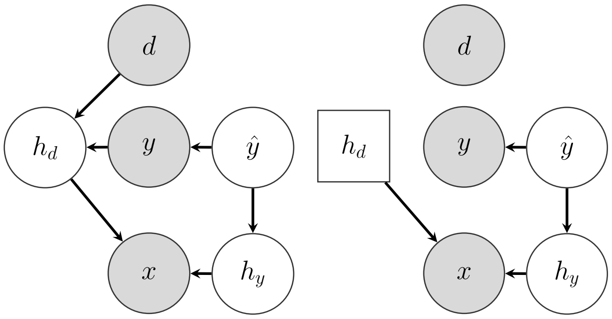

Figure 6. Left: DAG of the data generating process for the colored MNIST dataset. Right: The same DAG after intervention on h_d. Interventional nodes are squared.

A.3. PACS

Example images of the PACS dataset, see Figure 7

A.4. Linear example of intervention-augmentation equivariance

A simple linear example can be constructed where the domain d causes a specific ordering in h_d that is spuriously correlated with the label y. In addition, G is the permutation group and $g \in G$ acts as a permutation matrix A on x, i.e., $Ax = g \cdot x$. In particular, we assume that $f_X(\cdot)$ is a linear transformation

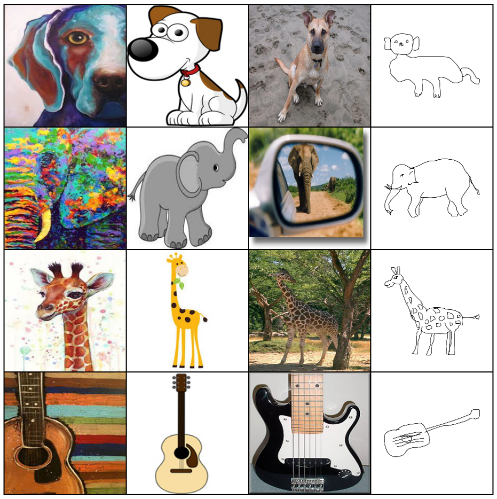

Figure 7. Samples from the first four classes ('dog', 'elephant', 'giraffe', 'guitar') for each domain (art-painting (A), cartoon (C), photo (P), sketch (S)) of the PACS dataset (Li et al., 2017a).

$$x = f_X(h_d, h_y) = Ch_d + Dh_y + e, \qquad (10)$$

where x, h_d, h_y, e are vectors and C, D are matrices correspondingly sized. The data augmentation can be expressed as a linear transformation of the form

$$x_{\text{aug}} = \text{aug}_A(x) = Ax, \qquad (11)$$

where A is a correspondingly sized matrix sampled from the set of all permutation matrices. Combining Equation 10 and 11, we obtain

$$\begin{aligned} x_{\text{aug}} &= Ax \\ &= ACh_d + ADh_y + Ae \\ &= C(C^{-1}ACh_d) + ADh_y + Ae \\ &= f_X(\text{do}_A(h_d), h_y). \end{aligned} \qquad (12)$$

We find that if that $AD = D$ and $Ae = e$, i.e., D and e are permutation invariant, the transformation $Ax = g \cdot x$

successfully simulates the noise intervention $\mathrm{do}_A(h_d) := C^{-1}ACh_d$ (with slight abuse of notation), i.e., we find that it satisfy the intervention-augmentation equivariance condition.

A.5. Causality

What follows is a brief introduction of causal concepts that are used throughout this paper. It hopefully makes the paper more self-contained, as well as more accessible for readers that encounter these concepts for the first time. For an in-depth introduction please see Pearl (2009) or Peters et al. (2017).

A.5.1. Structural causal models

We say that a set of variables $x_1, \dots, x_l$ causes a variable y if *intervening* on any of the x_m changes the distribution of y. This is usually different from (conditional) *observational* dependence between the x_m and y. Structural Causal Models (SCMs) are used to formalize those causal interactions between variables. We need to distinguish between two types of variables: exogenous and endogenous variables. Exogenous variables can be seen as an entry point to our SCM (and are usually unobserved independent random variables). The endogenous variables x_m are then determined by the causal mechanisms, which are formalized via functional relations: $x_m = f_m(x_{\mathrm{pa}_m})$, where x_{pa_m} is the tuple of the so-called parent variables of x_m. These relations of an SCM induces a corresponding graphical model. In this paper, we only deal with acyclic relationships, leading to Directed Acyclic Graphs (DAGs) as part of a Bayesian network. In Figure 8, we see three SCMs and their corresponding DAGs. Note that the direction of the arrows indicates the causal direction.

The SCMs in Figure 8 are considered to be the three main building blocks of every causal model: chain, confounder, and collider. Where each of them introduces a different (conditional in-)dependence structure. First row: In case of a chain the variables x and z become conditionally independent if we condition on the center variable y, i.e., $p(z|x, y) = p(z|y)$. Second row: An observed confounder y can introduce spurious correlation between its two children variables x and z, i.e., we may have $p(x, z) \neq p(x)p(z)$. If we condition on the confounding variable y they become conditionally independent again, i.e., $p(z|x, y) = p(z|y)$ and $p(x|z, y) = p(x|y)$. Third row: In case of an unobserved collider y the two parent variables are independent, $p(x, z) = p(x)p(z)$. However, if we condition on y they may become conditionally dependent, i.e., $p(x, z|y) \neq p(x|y)p(z|y)$.

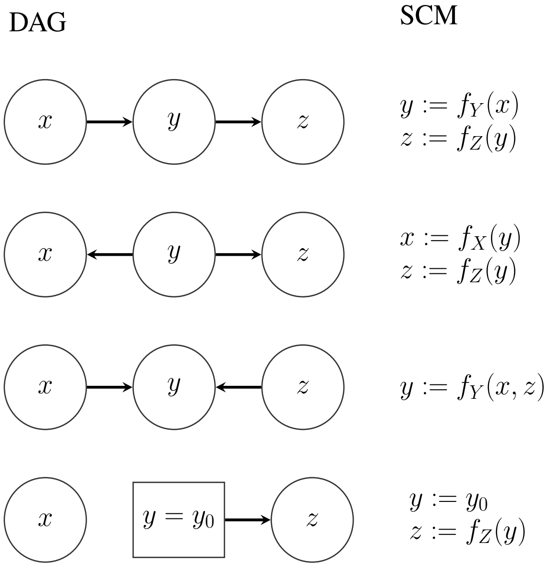

Figure 8. Top to bottom: chain, confounder, collider, chain with intervention on y.

A.5.2. Interventions

In its simplest form an intervention can be described as setting a variable y to a constant value, e.g., $y = y_0$ irrespective of its parent variables. The result of such an intervention on the SCM of a chain and the corresponding DAG can be seen in the bottom row of Figure 8. In this example, the variable y becomes independent of its parent variable x, i.e., we are replacing the function assignment $y = f(x)$ with $y = y_0$, effectively deleting the function $f(\cdot)$ and the corresponding arrow in the DAG. Using the *do*-operator (Pearl, 2009) we can write the resulting interventional distribution as follows: $p(z|x, \mathrm{do}(y = y_0)) = p(z|\mathrm{do}(y = y_0))$. In this paper, we use a special form of interventions, so-called noise or stochastic interventions (Peters et al., 2016). Instead of setting the intervened variable to a fixed value, we randomize the values of y, i.e., $\mathrm{do}(y = \xi)$, where ξ is sampled from a noise distribution N_ξ.

A.6. Domain generalization

A.6.1. Domain generalization via invariant feature representation

Arguably, the most commonly used approach in domain generalization relies on learning domain invariant features. The learning of invariant features can be achieved by mapping an input x to intermediate features z that are uninformative of the domain d, i.e., $p(z|d = i) = p(z|d = j)$. At the same time, the intermediate features z are optimized for a low prediction error on all training domains. This results in finding a saddle point for the setting commonly referred to as domain adversarial learning (Ganin et al., 2016). It is assumed that such z will generalize well to the test domain and, thus, result in a low test error.

Recent work of Zhao et al. (2019), Johansson et al. (2019) and Arjovsky et al. (2019), in the context of domain adaptation, shows that enforcing $p(z|d=i) = p(z|d=j)$ is not necessarily leading to a low test error if the domains d and targets y are spuriously correlated, i.e., $p(y|d=i) \neq p(y|d=j)$. We now extend the findings of Zhao et al. (2019) to domain generalization.

As shown in Zhao et al. (2019) an information-theoretic lower bound can be derived for the domain adaptation case. The bound "demonstrates that learning invariant representations could lead to a feature space where the joint error on both domains is large." We provide a straightforward extension of this bound for the domain generalization case.

Introduction of notation:

- $\mathbf{x}$: input
- $\mathbf{z}$: intermediate representation
- $\hat{y}$: output
- function composition: $x \xrightarrow{g} z \xrightarrow{h} \hat{y}$
- y: true label
- h: function mapping x to z
- g: function mapping z to $\hat{y}$
- JSD: Jensen-Shannon divergence
- $\epsilon^{d=i}$: empirical risk on domain $d = i$

Besides, we need the following two lemmas from Zhao et al. (2019). Proofs can be found in Zhao et al. (2019).

Lemma 4.6:

$$\begin{aligned} &\mathrm{JSD}(p(\hat{y}|d=i)||p(\hat{y}|d=j)) && (13)\\ &\leq \mathrm{JSD}(p(z|d=i)||p(z|d=j)), && (14)\end{aligned}$$

where $p(\hat{y}|d=i)$ are the marginal distributions of the output in domain $d=i$ and $p(z|d=i)$ are the marginal distributions of the intermediate representation in domain $d=i$.

Lemma 4.7:

$$\mathrm{JSD}(p(y|d=i)||p(\hat{y}|d=i)) \leq \sqrt{\epsilon_i(h \circ g)}, \tag{15}$$

i.e., how well is my output distribution matching the true labels distribution.

We start with the pairwise sum of Jensen-Shannon divergence between all N training domains and the $N+1$ test domain

$$\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(y|d=i)||p(y|d=j)). \tag{16}$$

Since JSD is a metric we can write

$$\begin{aligned} &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(y|d=i)||p(y|d=j)) && (17)\\ \leq &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(\hat{y}|d=i)||p(\hat{y}|d=j)) && (18)\\ &+2\sum_{k}^{N+1} \mathrm{JSD}(p(y|d=k)||p(\hat{y}|d=k)). && (19)\end{aligned}$$

Using Lemma 4.6 we get

$$\begin{aligned} &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(y|d=i)||p(y|d=j)) && (20)\\ \leq &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(z|d=i)||p(z|d=j)) && (21)\\ &+2\sum_{k}^{N+1} \mathrm{JSD}(p(y|d=k)||p(\hat{y}|d=k)). && (22)\end{aligned}$$

Using Lemma 4.7 we get

$$\begin{aligned} &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(y|d=i)||p(y|d=j)) && (23)\\ \leq &\sum_{1 \leq i < j \leq N+1} \mathrm{JSD}(p(z|d=i)||p(z|d=j)) && (24)\\ &+2\sum_{k}^{N+1} \sqrt{\epsilon^{d=k}(h \circ g)}. && (25)\end{aligned}$$

Extracting terms that belong to the test domain $d = N+1$ leads to

$$\begin{aligned} &\sum_{l=1}^{N} \mathrm{JSD}(p(y|d=l)||p(y|d=N+1)) && (26)\\ &+ \sum_{1 \leq i < j \leq N} \mathrm{JSD}(p(y|d=i)||p(y|d=j)) && (27)\\ &\leq \sum_{l=1}^{N} \mathrm{JSD}(p(z|d=l)||p(z|d=N+1)) && (28)\\ &+ \sum_{1 \leq i < j \leq N} \mathrm{JSD}(p(z|d=i)||p(z|d=j)) && (29)\\ &+2\sqrt{\epsilon^{d=N+1}(h \circ g)} + 2\sum_{k}^{N} \sqrt{\epsilon^{d=k}(h \circ g)} && (30)\end{aligned}$$

Assuming we find a perfect intermediate representation z for all N training domains and the test domain $d = N + 1$ (assuming such an z exists) we are left with

$$\sum_{l=1}^{N} \text{JSD}(p(y|d = l)||p(y|d = N + 1)) \tag{31}$$

$$+ \sum_{1 \leq i < j \leq N} \text{JSD}(p(y|d = i)||p(y|d = j)) \tag{32}$$

$$\leq 2\sqrt{\epsilon^{d=N+1}(h \circ g)} + 2\sum_{k}^{N} \sqrt{\epsilon^{d=k}(h \circ g)} \tag{33}$$

We see that, as it was the case for domain adaptation, that the joint risk across all domains (training and test) is lower bounded by the pairwise divergence of the marginal label distribution of all domains. Given the existence of an unobserved confounder as seen in Figure 1 the marginal label distribution are unlikely to match.

However, there exists a multitude of domain generalization methods that do not explicitly address the problem of hidden confounders (Balaji et al., 2018; Carlucci et al., 2018; 2019; Ding & Fu, 2018; Ghifary et al., 2015; Ilse et al., 2019; Li et al., 2017b; Mancini et al., 2018; Motiian et al., 2017; Shankar et al., 2018; Tzeng et al., 2014; Wang et al., 2018). However, the majority of these methods are evaluate on benchmark datasets, e.g., VLCS (Khosla et al., 2012) or PACS (Li et al., 2017a), where the domain d and the target y are confounded. As shown in Equation 33, this can result in poor generalization performance. Nonetheless, we cannot rule out the possibility that some of these methods are implicitly able to deal with confounders, thus achieving good generalization performance.

To the best of our knowledge, there are currently very few methods that address the issue of spuriously correlated domains d and targets y (Arjovsky et al., 2019; Heinze-Deml & Meinshausen, 2019; Li et al., 2018; Krueger et al., 2020), where Li et al. (2018) extends the idea of domain adversarial learning to enforce conditional domain invariance, i.e., $p(z|y, d = i) = p(z|y, d = j)$.

A.7. Data augmentation

We will briefly summarize how data augmentation is currently viewed in the computer vision community, for a in-depth survey see Shorten & Khoshgoftaar (2019). In computer vision data augmentation is seen as an effective technique for improving the performance on a variety of tasks such as image classification, object detection, and image segmentation. In the image domain, data augmentation techniques can be roughly divided into two categories:

1. Augmenting the geometry of an image: Commonly used transformations are rotations, horizontal and vertical flips, scaling, cropping, occlusion, and elastic deformations.
2. Augmenting the color of an image: Random values are added or subtracted from the color channels of an image. Instead of applying this transformation directly in the RGB colorspace, other color spaces like CIELAB and HSL are commonly used (Tellez et al., 2019).

Data augmentation is a combination of the transformation listed above that are randomly applied to all images during training.

A.7.1. Data augmentation in application-focused research areas

In the following, we give a summary of two examples of the successful application of data augmentation for domain generalization in medical imaging and robotics. We want to highlight that in both examples the actual task and the domains are spuriously correlated.

Histopathology The high variability of the appearance of histopathology images is a major obstacle for the deployment of automatic image analysis systems. The variability of appearance is the result of a multitude of preparation steps that are applied to the specimen: cutting, fixating, staining, and scanning. Each step introduces its own artifacts. This leads to different color distributions among histopathology laboratories. Tellez et al. (2019) perform a detailed comparison of commonly used data augmentation, see Appendix Figure 9. The augmentation techniques consist of random rotation and flipping, random color perturbation, and color normalization. These augmentation techniques are compared on a dataset composed of histopathology images from nine different laboratories. We argue that there exists a hidden confounder that spuriously correlates the staining and scanner artifacts (caused by the laboratories) and the abnormalities in the tissue (caused by the diseases). By augmenting the color of the histopathology images Tellez et al. (2019) are able to learn features that are invariant to the laboratories. Furthermore, Tellez et al. (2019) find that random color perturbation outperforms color normalization. We argue that random color perturbation simulates noise interventions, whereas color normalization tries to simulate interventions where the color of a histopathology image is set to a fixed value. As described in Section 2, this requires to first estimate the color distribution of the original histopathology image which is a challenging problem. As a result, data augmentation in the form of random color perturbation is better suited to simulate interventional data.

Robotics Performing robotic learning on physical hardware is often not feasible due to: (*i*) the large number of training samples that are required, and (*ii*) potential damage

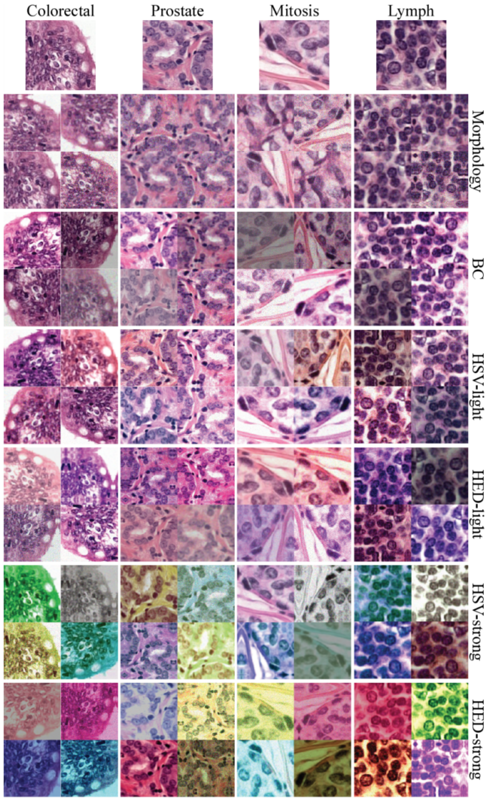

Figure 9. Domain randomization histopathology, taken from Tellez et al. (2019)

to the hardware if the learning relies on random exploration. Therefore, learning in a physics simulator is of great interest. While learning in a simulator is cheap and safe, we are facing a new problem, namely, how to overcome the so-called *reality gap*, i.e., the differences between simulation and the real world. In Tobin et al. (2017) they focus on a robotic manipulation task that involves a robotic arm and eight 3D objects that are placed on a table. In this scenario, a neural network is used to detect the location of an object. To be able to generalize from the simulation to the real world, Tobin et al. (2017) apply a variety of data augmentation techniques to the simulator, e.g., randomization of position and texture of all objects on the table, textures of the table, floor, skybox, and robot, and the addition of random noise. We argue that there exists a hidden confounder that introduces a spurious correlation between, e.g., the lighting conditions and the location of the objects on the table. By applying heavy data augmentation during the training process they are able to generalize to unseen lighting conditions in the real world.

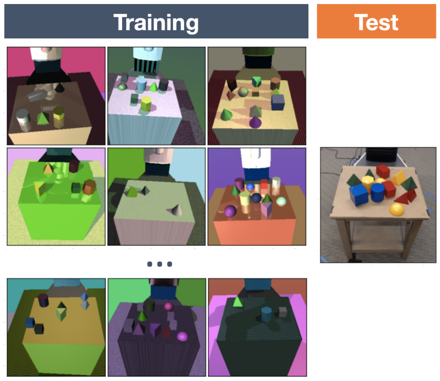

Figure 10. Domain randomization in robotics, taken from Tobin et al. (2017)