Selecting Data Augmentation for Simulating Interventions

Ilse, M.; Tomczak, J.M.; Forré, P.

Publication date
2021
Document Version
Final published version
Published in
Proceedings of Machine Learning Research
License
Other

Selecting Data Augmentation for Simulating Interventions

Maximilian Ilse [1] **Jakub M. Tomczak** [2] **Patrick Forré** [1]

Abstract

Machine learning models trained with purely observational data and the principle of empirical risk minimization (Vapnik, 1992) can fail to generalize to unseen domains. In this paper, we focus on the case where the problem arises through spurious correlation between the observed domains and the actual task labels. We find that many domain generalization methods do not explicitly take this spurious correlation into account. Instead, especially in more application-oriented research areas like medical imaging or robotics, data augmentation techniques that are based on heuristics are used to learn domain invariant features. To bridge the gap between theory and practice, we develop a causal perspective on the problem of domain generalization. We argue that causal concepts can be used to explain the success of data augmentation by describing how they can weaken the spurious correlation between the observed domains and the task labels. We demonstrate that data augmentation can serve as a tool for simulating interventional data. We use these theoretical insights to derive a simple algorithm that is able to select data augmentation techniques that will lead to better domain generalization.

1. Introduction

Despite recent advancements in machine learning fueled by deep learning, studies like Azulay & Weiss (2019) have shown that deep learning methods may not generalize to inputs from outside of their training distribution. In safety-critical fields like medical imaging, robotics and, self-driving cars, however, it is essential that machine learning models are robust to changes in the environment. Without the ability to generalize, machine learning models cannot be safely deployed in the real world.

*Equal contribution [1]Amsterdam Machine Learning Lab, University of Amsterdam [2]Computational Intelligence Group, Vrije Universiteit Amsterdam. Correspondence to: Maximilian Ilse <m.ilse@uva.nl>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

In the field of domain generalization, one tries to find a representation that generalizes across different environments, called *domains*, each with a different shift of the input. This problem is especially challenging when changes in the domain are spuriously associated with changes in the actual task labels. This can, for instance, happen when the data gathering process is biased. An example is given by Arjovsky et al. (2019): If we consider a dataset of images of cows and camels in their natural habitat, then there is a strong correlation between the type of animal and the landscape in the image, e.g., a camel standing in a desert. If we now train a machine learning model to predict the animal in a given image, the model is prone to exploit the spurious correlation between the type of animal and the type of landscape. As a result, the model can fail to recognize a camel standing in a green pasture or a cow standing in a desert.

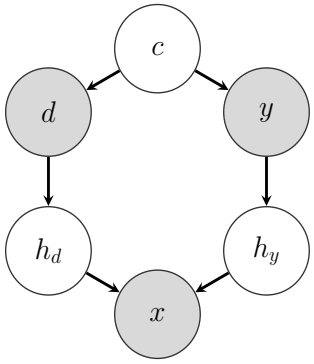
In recent years, a large corpus of methods designed to learn representations that will generalize across domains has been formulated. While the proposed methods are able to achieve good results on a variety of domain generalization benchmarks, the majority of them lack a theoretical foundation. In the worst-case scenario, these methods enforce the wrong type of invariance, as proven in Appendix A.6.1 . Interestingly, we find that especially in more applied fields, like medical imaging and robotics, researchers have found a practical way of dealing with the spurious correlation between domains and the actual task. Data augmentation in combination with Empirical Risk Minimization (ERM) (Vapnik, 1992) is used to enforce invariance of the machine learning model with respect to changes in the domain. Hereby, prior knowledge is used to guide the selection of appropriate data augmentation. In Appendix A.7.1, we give a detailed summary of two successful applications of data augmentation in the context of domain generalization.

However, the success of data augmentation is often described in vague terms like 'artificially expanding labeled training datasets' (Li, 2020) and 'reduce overfitting' (Krizhevsky et al., 2012). In this paper, we present a causal perspective on data augmentation in the context of domain generalization and contribute to the field in the following manner:

- First, we introduce the concept of *intervention-augmentation equivariance* that formalizes the rela-

tionship between data augmentation and interventions on features caused by the domain. We show that if intervention-augmentation equivariance holds we can use data augmentation to successfully simulate interventions using only observational data.

- Second, we derive a simple algorithm that is able to select data augmentation techniques from a given list of transformations. We compare our approach to a variety of domain generalization methods on three domain generalization benchmarks. We demonstrate that we are able to consistently outperform all other methods.



$$
\begin{aligned}
d &:= f_D(c) \\
y &:= f_Y(c) \\
h_d &:= f_{H_d}(d) \\
h_y &:= f_{H_y}(y) \\
x &:= f_X(h_d, h_y), \quad (1)
\end{aligned}
$$

Figure 1. DAG and SCM with a hidden confounder.

2. Method

2.1. Domain generalization

We first formalize the problem of domain generalization following the notations used in Muandet et al. (2013). We assume that during training we have access to samples $\mathcal{S}$ from N different domains, where $\mathcal{S} = \{S^{d=i}\}_{i=1}^{N}$. From each domain n_i samples $S^{d=i} = \{(x_k^{d=i}, y_k^{d=i})\}_{k=1}^{n_i}$ are included in the training set. The training data is represented as tuples of the form (x, y, d) sampled from the observational distribution $p(x, y, d)$. The goal of domain generalization is to develop machine learning methods that generalize well to unseen domains. In order to test the ability of a machine learning model to generalize, we use samples $S^{d=N+1}$ from a previously unseen test domain $d = N + 1$.

In this paper, we are interested in the general case where the observed domains d and targets y are spuriously correlated in the training dataset, i.e., where we might have $p(y|d = i) \neq p(y|d = j), i, j \in \{1, \dots, N\}$. Since the correlation between d and y is assumed to be spurious, it does not necessarily hold for the test domain $d = N + 1$.

2.2. Domain generalization and data augmentation from a causal perspective

For readers unfamiliar with the concepts of causality, a brief introduction of the causal concepts that are used throughout this paper can be found in Appendix A.5. For an in-depth introduction please see Pearl (2009) or Peters et al. (2017).

First, we introduce a Structural Causal Model (SCM) in order to describe what we believe in many cases reflects the underlying causal structure of domain generalization problems. The SCM is shown in Figure 1 (right) where c is a hidden confounder (and a exogenous variable), d the domain, y the target, h_d high-level features, e.g., color and orientation, caused by d, h_y high level-features, e.g., shape and texture, caused by y, and x the input. We omit including noise variables for clarity. The corresponding Directed Acyclic Graph (DAG) is shown in Figure 1 (left), where a grey node means the variable is observed and a white node corresponds to a latent (unobserved) variable. The presented DAG is similar to the ones constructed in Subbaswamy & Saria (2019) and Castro et al. (2019). In Figure 1, the node c is a hidden confounder. The hidden confounder c opens up a backdoor path (a non-causal path) $d \leftarrow c \rightarrow y$ (Pearl, 2009). This path allows d to enter y trough the back door.

As a result, the domain d and the target y are in general no longer independent, $p(y, d) \neq p(y)p(d)$. Since the high-level features, h_d are children of d, they are spuriously correlated with y as well, i.e., h_d becomes predictive of y. We now assume that we train a machine learning model using ERM (Vapnik, 1992) and observational data generated from the DAG in Figure 1. The task is to predict y from x, which itself is anti-causal. Since d and y are correlated, it is likely that the machine learning model will rely on all high-level features h_d and h_y to predict y. Furthermore, we assume that the correlation of d and y is spurious. Therefore, it will not hold in general and will break under intervention. A machine learning model relying on high-level features h_d that are caused by d is thus likely to generalize poorly to unseen domains. Returning to our introductory example of classifying animals in images, the hidden confounder can be used to model the fact that there is a common cause for the type of animal and the landscape in an image. For example, the confounder could be the country in which a particular image was taken, e.g., in Switzerland we are more likely to see a cow standing in a green pasture than a camel or a desert.

2.3. Simulating interventions

One possible approach to deal with the spurious correlations between d and y is to perform an intervention on d. Such an intervention would render d and y independent, i.e., $p(y|\text{do}(d)) = p(y)$. In Figure 2 (left), we see the same DAG as in Figure 1 but after we intervened on d. We find that in Figure 2 (left) there is no more arrow connecting the hidden confounder c and the domain d. The backdoor path

$d \leftarrow c \rightarrow y$ has vanished. In the examples of animals and landscapes, to intervene on the domain d (the landscape), we would have to physically move a cow to a desert. It becomes apparent that the interventions have to happen in the real world and are not operations on the already gathered observational data. In the majority of domain generalization problems, it will not be feasible to collect new data with specific interventions.

In Figure 2 (center) we present a second way of addressing the problem of correlated variables d and y. In theory one could perform an intervention on all high-level features h_d, i.e., $\mathrm{do}(h_d)$, since d affects x only indirectly via h_d, in our example h_d could represent the colors and textures of the landscapes. Again, an intervention like this would need to happen during the data collection process in the real world, e.g., by moving sand to a pasture.

However, we argue that in certain cases we can simulate data from the interventional distribution $p(x, y|\mathrm{do}(h_d))$ using data augmentation in combination with observational data. For example, we could randomly perturb the colors in the animal images. This type of augmentation simulates a noise intervention on h_d, i.e., $\mathrm{do}(h_d = \xi)$, where ξ is sampled from a noise distribution N_ξ (Peters et al., 2016).

In theory, we could intervene on h_d by setting h_d to a fixed value, instead of performing a noise intervention. However, in order to simulate data from such an interventional distribution using data augmentation, we would require h_d to be observed, which we argue is generally not the case. In Appendix A.7.1, we describe that there exist data augmentation methods that try to infer h_d for each sample x before setting h_d to a fixed value for all samples, yet these augmentations seem to perform worse than randomly sampled augmentations.

By augmenting only high-level features h_d that are caused by d we guarantee that the target y and features h_y are unchanged. After data augmentation the pairs (x_{aug}, y) should closely resemble samples from the interventional distribution $p(x, y|\mathrm{do}(h_d))$. In Figure 2 (right) we see that we only require observational data from the DAG without any interventions. While each augmented sample x_{aug} individually can be seen as a counterfactual, we argue that we effectively marginalize over the counterfactual distribution by generating a multitude of augmented samples x_{aug} from each x. We argue that for correctly chosen data augmentation we cannot distinguish the data generated by any of the three models in Figure 2.

If we want to choose data augmentation $x_{\text{aug}} = \mathrm{aug}(x)$, as a transformation $\mathrm{aug}(\cdot)$ applied to observed data x, such that it simulates an intervention on the high-level features h_d caused by d, one needs to make assumption about the causal data generating process. Formally, we require that augmenting the data x to $x_{\text{aug}} = \mathrm{aug}(x)$ commutes with an intervention $\mathrm{do}(h_d)$ prior to the data generation. We call this *intervention-augmentation equivariance*. In more formal detail, assume that we have the causal process from Equation 1: $x := f_X(h_d, h_y)$. Then augmenting x via $\mathrm{aug}(\cdot)$ does:

$$\begin{aligned} x_{\text{aug}} &= \mathrm{aug}(x) \\ &= \mathrm{aug}(f_X(h_d, h_y)). \end{aligned} \tag{2}$$

We then say that the causal process $f_X : \mathcal{H}_d \times \mathcal{H}_y \mapsto \mathcal{X}$, is *intervention-augmentation equivariant* if for every considered stochastic data augmentation transformation $\mathrm{aug}(\cdot)$ on $x \in \mathcal{X}$ we have a corresponding noise intervention $\mathrm{do}(\cdot)$ on $h_d \in \mathcal{H}_d$ such that:

$$\mathrm{aug}(f_X(h_d, h_y)) = f_X(\mathrm{do}(h_d), h_y). \tag{3}$$

The intervention-augmentation equivariance is expressed as a commutative diagram in Figure 3. We argue that by making strong assumptions about the true causal process we need to first identify the high-level features h_d caused by d. Second, we have to choose data augmentation $\mathrm{aug}(x)$ that commutes with a corresponding intervention $\mathrm{do}(h_d)$ under the causal process $f_X(h_d, h_y)$. A special case of intervention-augmentation equivariance occurs in the classical case of an G-equivariant map f_X, where G can be any (semi-)group. For this to hold, we need G to act on the spaces $\mathcal{H}_y$, $\mathcal{H}_d$, $\mathcal{X}$, and we need to make sure that G acts trivially on $\mathcal{H}_y$. So any element $g \in G$ can transform elements $x \in \mathcal{X}$ into $g \cdot x \in \mathcal{X}$, which we will interpret as data augmentation, as demonstrated in Section 4. The elements $g \in G$ also transform $h_d \in \mathcal{H}_d$ into $g \cdot h_d \in \mathcal{H}_d$, which we consider as a special type of intervention. Furthermore, $h_y \in \mathcal{H}_y$ are assumed to be kept fixed $g \cdot h_y = h_y$ for all $g \in G$. So we put:

$$\mathrm{do}(h_d) := g \cdot h_d, \tag{4}$$

$$\mathrm{aug}(x) := g \cdot x, \tag{5}$$

where we assume that the elements $g \in G$ are randomly sampled from some distribution $p(g)$ on G. In this setting, any *G-equivariant map* f_X is then automatically also intervention-augmentation equivariant, as can be seen from:

$$\mathrm{aug}(x) = g \cdot f_X(h_d, h_y) \tag{6}$$

$$= f_X(g \cdot h_d, g \cdot h_y) \tag{7}$$

$$= f_X(\mathrm{do}(h_d), h_y), \tag{8}$$

a linear example of intervention-augmentation equivariance can be found in the Appendix.

In general, we find that the majority of frequently used data augmentations can be expressed as simple group actions. For example, randomly rotating the input image x can be

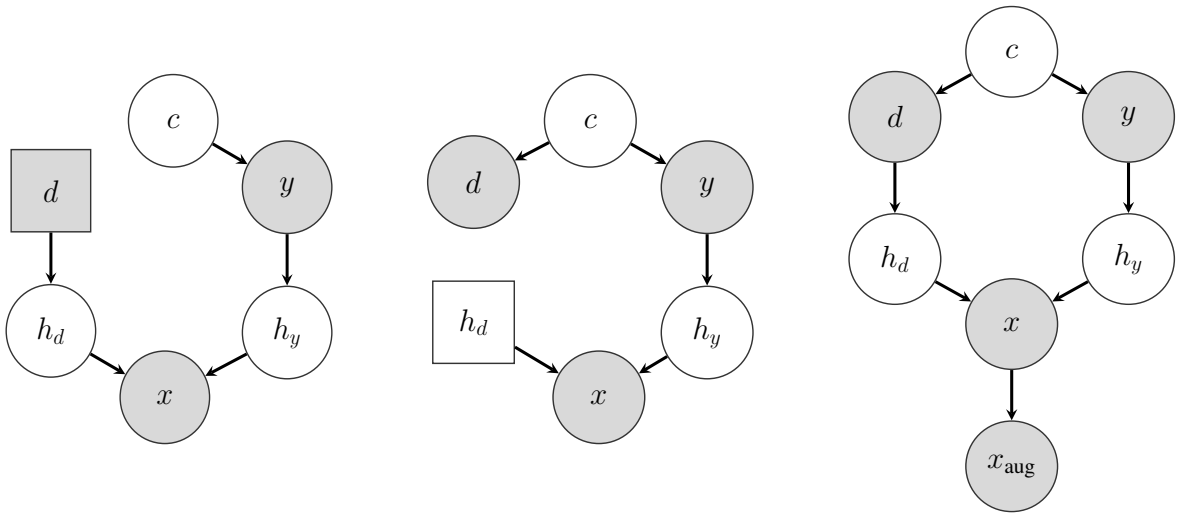


Figure 2. Left: DAG with hidden confounder after intervention on d. Center: DAG with hidden confounder after intervention on h_d. Interventional nodes are squared. Right: DAG with hidden confounder plus data augmentation. Note that in the latter case we do not have to intervene on the system that generates the data. Data augmentation should be chosen in a way such that the augmented data simulates data from the center or left DAG.

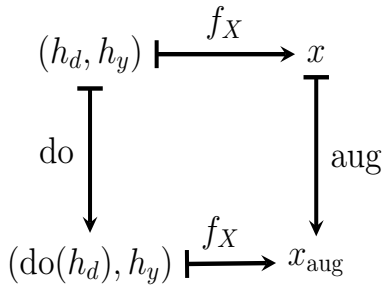


Figure 3. Intervention-augmentation equivariance expressed in a commutative diagram.

understood as randomly sampling and applying elements g from the two-dimensional rotation group $SO(2)$ on the two dimensional pixel grid. Randomly changing the hue of an image x corresponds to randomly sampling and applying elements g from the two-dimensional rotation group $SO(2)$, since hue can be represented as an angle in color space. Applying random permutations to the color channels of an image x is equivalent to randomly sampling and applying elements g from permutation group S_3, in the case of three separate color channels.

2.4. Selecting data augmentations for domain generalization

In Figure 2 (center), we see that if we successfully simulate an intervention on h_d using data augmentation the arrow from d to h_d vanishes. Based on this theoretical insight, we propose an algorithm that is able to select data augmentation techniques that will improve domain generalization, instead of manually choosing them. In the following we will refer to the algorithm as Select Data Augmentation (SDA). Similar to Cubuk et al. (2019), we start with a list of data augmentation techniques including: 'brightness', 'contrast', 'saturation', 'hue', 'rotation', 'translate', 'scale', 'shear', 'vertical flip', and 'horizontal flip'. Since these transformations do not influence each other, they can be tested separately. The hyperparameter for each augmentation can be found in the Appendix. The proposed SDA algorithm consists of three steps:

1. We divide all samples from the training domains into a training and validation set.
2. We train a classifier to predict the domain d from input x. During training, we apply the first data augmentation in our list to the samples of the training set. We save the domain accuracy on the validation set after training. We repeat this step with all data augmentations in the list.
3. We select the data augmentation with the lowest domain accuracy averaged over five seeds. If multiple data augmentations lie within the standard error of the selected one they are selected as well, i.e., there is no statistically significant difference between the augmentations.

Intuitively, SDA will select data augmentation techniques that destroy information about d in x. From a causal point of view, this is equivalent to weaken the arrow from d to h_d. In Appendix A.1.1, we perform an ablation study showing

that SDA also reliably selects the most suitable data augmentation if the list contains the same augmentation with different hyperparameters.

There is one caveat though. Throughout this entire section, we assume that we are successfully augmenting all high-level features h_d caused by d. In a real-world application, we usually have no means to validate this assumption, i.e., we might only augment a subset of h_d. Furthermore, we might even augment high-level features h_y that are caused by the target node y. Nonetheless, we argue there are cases where we still obtain better generalization performance than a machine learning model trained without data augmentation. This may happen in cases where weakening the spurious confounding influence of h_d on y recovers more of the anti-causal signal for y than the data augmentation on the features h_y destroys. We evaluate this hypothesis empirically in Section 4.

3. Related work

3.1. Learning symmetries from data

In the previous section, we argue that choosing the right symmetry group for data augmentation relies on prior knowledge, e.g., preselecting a list of transformations to test. While this is a clear practical limitation of our approach, to the best of our knowledge there exist no approaches that are able to learn symmetries from purely observational data. Contemporary approaches like Lagrangian neural networks (Cranmer et al., 2020), graph neural networks (Kipf & Welling, 2017), and group equivariant neural networks (Cohen & Welling, 2016) are enforcing apriori chosen symmetries instead of learning them.

3.2. Understanding data augmentation

Recently, Gontijo-Lopes et al. (2020) develop two measures: affinity and diversity. The measures are used to quantify the effectiveness of existing data augmentation methods. They find that augmentations that have high affinity and diversity scores lead to better generalization performance. While affinity and diversity rely on the iid assumption, we provide an alternative for non-iid datasets. Lyle et al. (2020) investigate how data augmentation can be used to incorporate invariance into machine learning models. They show that while data augmentation can lead to tighter PAC-Bayes bounds, data augmentation is not guaranteed to lead to invariance. In Equation, 3 we formalize under which condition (namely intervention-augmentation equivariance) data augmentation will lead to invariance.

3.3. Advanced data augmentation techniques

Zhang et al. (2018) introduced a method called mixup that constructs new training examples by linearly interpolating between two existing examples (x_i, y_i) and (x_j, y_j). In Gowal et al. (2019) and Perez & Wang (2017) a Generative Adversarial Network (GAN) is used to perform so-called 'adversarial mixing'. The GAN is able to generate new training examples that belong to the same class y but have different styles. Furthermore, Perez & Wang (2017) propose a novel method called 'neural augmentation' where they train the first part of their model to generate an augmented image from two training examples with the same class y.

3.4. Causality

In Peters et al. (2016) a method for Invariant Causal Prediction (ICP) is developed. It is built on the assumption that causal features are stable given different experimental settings. Given the complete set of causal features, the conditional distribution of the target variable y must remain the same under interventions, e.g., change of the domain. Whereas, predictions made by a machine learning model relying on non-causal features are in general not stable under interventions. Recently, Arjovsky et al. (2019) proposed a framework called Invariant Risk Minimization (IRM), that shares the same goal as ICP. In IRM a soft penalty in combination with an ERM term is used to balance the invariance and predictive power of the learned machine learning model. In contrast to ICP, IRM can be used for tasks on unstructured data, e.g., images. However, while both methods (ICP and IRM) try to learn features that are parents of y, we argue that for the majority of domain generalization problems the task of predicting y from x is anti-causal. Therefore we are interested in augmenting only features caused by d, i.e., the descendants of d, assuming that the remaining features are caused by y. In Arjovsky et al. (2019), they argue that there exists a discrepancy between the true label (part of the true causal mechanism) that caused x and the annotation produced by human labelers. Learning this 'labeler function' will lead to a good generalization performance, even though it might rely on patterns that are anti-causal or non-causal. In this situation, the IRM objective becomes ineffective.

Heinze-Deml & Meinshausen (2019) introduced the Conditional variance Regularization (CoRe). CoRe uses grouped observations (e.g., training samples with the same class y but different styles) to learn invariant representations. Samples are grouped by an additional ID variable, which is different from the label y. We find that in most cases it is difficult to obtain an additional ID variable, e.g., none of the datasets in Section 4 features such a variable. If no such ID variable exists, CoRe can use pairs of original images and augmented images to learn invariant representations.

While we are focusing on the DAG in Figure 1, Bareinboim

& Pearl (2016) and Mooij et al. (2019) have developed general graphical representations for relating data generating processes across domains. If the confounder c was observed methods that find stable feature sets such as those in Rojas-Carulla et al. (2018) and Magliacane et al. (2018), could be used. Furthermore, Subbaswamy et al. (2019) shows that instead of intervening in some cases, it is possible to fit an interventional distribution from observational data. However, imaging data poses a challenge that existing causal-based methods are not equipped to deal with thus motivating the use of data augmentation.

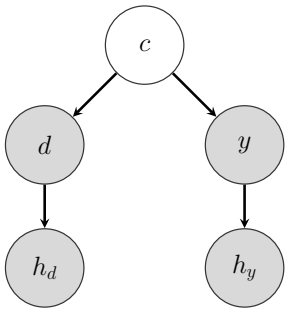
4. Experiments

We evaluate the performance of data augmentation in combination with Empirical Risk Minimization (ERM) (Vapnik, 1992) on four datasets. While the first is a synthetic dataset, the other three are domain generalization benchmark image datasets (rotated MNIST, colored MNIST, and PACS) where the domain d and target y are confounded. The synthetic dataset is used to study the effect of data augmentation on a model's performance when high level-features caused by domain as well as high level-features caused by the label are augmented. For the benchmark image datasets, we first use SDA to select the best data augmentation techniques. The results for this first step can be found in Table 5 in the Appendix. Afterwards, we apply the selected data augmentations and train the respective model using ERM. Finally, we perform an ablation study where we apply all data augmentations to all three image datasets instead of the selected ones.

Code to replicate all experiments can be found under `https://github.com/AMLab-Amsterdam/DataAugmentationInterventions`.

4.1. Synthetic data

For the first experiment we simulate data from the linear Gaussian SCM in Figure 4 (right), where the corresponding DAG can be seen in Figure 4 (left).



$$
\begin{aligned}
c &:= \mathcal{N}(0, \sigma_c^2) \\
d &:= c \cdot W_{c\to d} + \mathcal{N}(0, \sigma^2) \\
y &:= c \cdot W_{c\to y} + \mathcal{N}(0, \sigma^2) \\
h_d &:= d \cdot W_{d\to h_d} + \mathcal{N}(0, \sigma^2) \\
h_y &:= y \cdot W_{y\to h_y} + \mathcal{N}(0, \sigma^2),
\end{aligned} \tag{9}
$$

Figure 4. DAG and linear Gaussian SCM for synthetic data.

We choose c, d, y, h_d and h_y to be five dimensional vectors. Furthermore, we sample the elements of the square matrices $W_{c\to d}$, $W_{c\to y}$, $W_{d\to h_d}$ and $W_{y\to h_y}$ from $\mathcal{N}(0, I)$. In all of our experiments $\sigma_c = I$ and $\sigma = 0.1 \cdot I$. The task is to regress $\sum_i^5 y_i$ from x, where $x = [h_d, h_y]$, a 10 dimensional feature vector. During training the data is generated using the DAG in Figure 4 (left), where due the confounder c the features h_d and y are spuriously correlated. During testing we set $d := \mathcal{N}(0, I)$, keeping $W_{c\to d}$, $W_{c\to y}$, $W_{d\to h_d}$ and $W_{y\to h_y}$ the same as during training. As a result, features h_d and y are no longer correlated. A model relying on features h_d will not be able to generalize well to the test data. In all experiments, we use linear regression to minimize the empirical risk. We choose to add noise sampled from a uniform distribution $U[-10, 10]$ as our data augmentation technique. We vary the number of dimensions of h_d as well as of h_y that are augmented. Each experiment is repeated 50 times, in Figure 5 we plot the mean of the mean square error (MSE) together with the standard error.

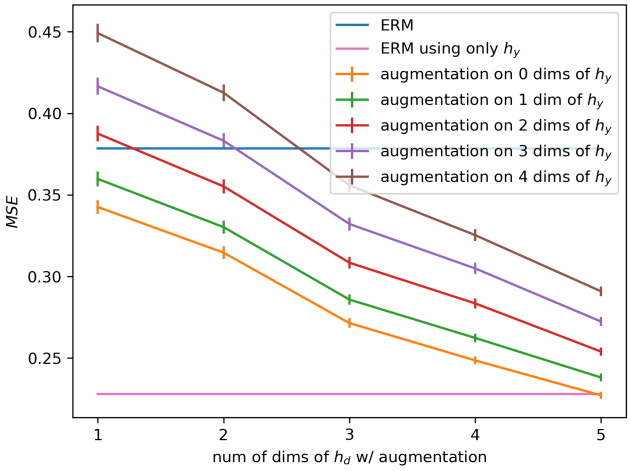


Figure 5. Results on synthetic data.

In Figure 5, we see that ERM using only features h_y (pink line) achieves the lowest MSE. Next, we apply data augmentation to one, two, three, four, and five dimensions of h_d while keeping h_y unchanged (orange line). We find that if data augmentation is applied to all five dimensions of h_d we can match the MSE of ERM with only features h_y. In this case, we are satisfying the condition in Equation 3. Furthermore, we find that unsurprisingly the MSE of models trained with data augmentation applied to features h_y increases (green, red, purple, and brown line). However, we can see that as long as we apply data augmentation to at least three dimensions of h_d the resulting MSE is lower than ERM using all features h_d and h_y (blue line). Perhaps the most surprising result of this experiment is that there exist conditions under which applying data augmentation to features caused by d *and* features caused by y will result in

better generalization performance compared to ERM using all features.

4.2. Rotated MNIST

We construct the rotated MNIST dataset following Li et al. (2018). This dataset consists of four different domains d and ten different classes y, each domain corresponds to a different rotation angle: $d = \{0°, 30°, 60°, 90°\}$. We first randomly select a subset of images x from the MNIST training dataset and afterward apply the rotation to each image of the subset. For the next domain, we randomly select a new subset. To guarantee the variance of $p(y)$ among the domains, the number of training examples for each digit class y is randomly chosen from a uniform distribution $U[80, 160]$.

For each experiment three of the domains are selected for training and one domain is selected for testing. For the test domain, the corresponding rotation is applied to the 10000 examples of the MNIST testset. In Table 2, we compare data augmentation in combination with ERM to ERM, a Domain Adversarial Neural Network (DANN) (Ganin et al., 2016) and a Conditional Domain Adversarial Neural Network (CDANN) (Li et al., 2018). All methods use a LeNet (LeCun et al., 1998) type architecture and we repeat each experiment 10 times. First, we use SDA to find the best data augmentation technique, where we use the same LeNet model and training procedure for the domain classifier and only samples from the training domains. The data augmentation with the lowest domain accuracy in all four cases, where we leave out one of the domains for testing, is 'rotation'. In addition, we perform an ablation study showing that SDA reliably picks the most suitable hyperparameters, the results can be found in Table 4 in the Appendix. Second, we apply random rotations between $0°$ and $359°$ to the images x during training, denoted by DA. If we assume h_d to be equal to the rotation angle of the MNIST digit in a given image x, applying random rotations to x is equal to a noise intervention on h_d, see Equation 3. As described in Section 2, applying random rotations to x can be understood as randomly sampling elements g from the two-dimensional rotation group $SO(2)$. Note that elements $g \in SO(2)$ act trivially on h_y: Rotations do not change the digit shapes. The result is a training dataset where d and y are independent. In Table 2, we see that the results of DA are similar for all four test domains. Furthermore, we find that DA outperforms ERM, DANN, and CDANN, where CDANN is specially designed for the case where d and y are spuriously correlated.

Table 1. Results on Colored MNIST. Average accuracy $\pm$ standard deviation for ten seeds.

Acc	ERM	IRM	REx	SDA
Train	**87.4 $\pm$ 0.2**	70.8 $\pm$ 0.9	71.5 $\pm$ 1.0	72.1 $\pm$ 0.4
Test	17.1 $\pm$ 0.6	66.9 $\pm$ 2.5	68.7 $\pm$ 0.9	**74.1 $\pm$ 0.9**

Table 2. Results on Rotated MNIST results. Average accuracy for ten seeds.

Target	ERM	DANN	CDANN	SDA
$0°$	75.4	77.1	78.5	**96.1**
$30°$	93.4	94.2	94.9	**95.9**
$60°$	94.5	95.2	95.6	**95.7**
$90°$	79.6	83.0	84.0	**95.9**
Ave	85.7	87.4	88.3	**95.9**

4.3. Colored MNIST

Following Arjovsky et al. (2019), we create a version of the MNIST dataset where the color of each digit is spuriously correlated with a binary label y. We construct two training domains and one test domain where the digits of the original MNIST classes '0' to '4' are labeled $y = 0$ and the digits of the classes '5' to '9' are are labeled $y = 1$. Subsequently, for 25% of the digits we flip the label y. We now color digits which are labeled $y = 0$ red and digits which are labeled $y = 1$ green. Last, we flip the color of a digit with a probability of 0.2 for the first training domain and with a probability of 0.1 for the second training domain. In the case of the test domain, the color of a digit is flipped with a probability of 0.9. By design, the original MNIST class of each digit ('0' to '9') is a direct cause of the new label y whereas the color of each digit is a descendant of the new label y.

The DAG of the colored MNIST, shown in Appendix Figure 6, deviates slightly from the DAG in Figure 1, nonetheless the reasoning in Section 2 is still valid. In Table 1, we see that while ERM is performing well on the training domains it fails to generalize to the test domain since it is using the color information to predict y. In contrast, IRM (Arjovsky et al., 2019) and REx (Krueger et al., 2020) generalizes well to the test domain. Again, we use SDA to find the appropriate data augmentations. We use the same MLP and training procedure as in Arjovsky et al. (2019) for the domain classifier. We want to highlight that SDA only relies on samples from the two training domains whereas the hyperparameters of IRM and REx where fine-tuned on samples from the test domain as described in Krueger et al. (2020). In case of the colored MNIST dataset the selected data augmentations are 'hue' and 'translate', denoted by DA. As described in the Section 2, applying random permutations to the hue value of x is equivalent to randomly sampling and applying elements g from permutation group $SO(2)$. We argue that elements g do not change h_y: high-level features that contain information about the shape of each digit. In our experiment, we use the same network architecture and training procedure as described in Arjovsky et al. (2019). Each experiment is repeated 10 times. We find that DA can successfully weaken the spurious confounding influence of the domain d on y, see Table 1.

Table 3. Results on PACS dataset. Average accuracy for five seeds.

Target	ERM	CDANN	L2G	GLCM	SSN	IRM	REx	MetaReg	JigSaw	SDA
A	63.3	62.7	66.2	66.8	64.1	67.1	67.0	69.8	67.6	**70.45**
C	63.1	69.7	66.9	69.7	66.8	68.5	68.0	70.4	**71.7**	68.49
P	87.7	78.7	88.0	87.9	90.2	89.4	89.7	**91.1**	89.0	88.35
S	54.1	64.5	59.0	56.3	60.1	57.8	59.8	59.3	65.2	**72.24**
Ave	67.1	68.9	70.0	70.2	70.3	70.7	71.1	72.6	73.4	**74.9**

4.4. PACS

The PACS dataset (Li et al., 2017a) was introduced as a strong benchmark dataset for domain generalization methods that features large domain shifts. The dataset consists of four domains: d = ['photo' (P), 'art-painting' (A), 'cartoon' (C), 'sketch' (S)], i.e., each image style is viewed as a domain. The numbers of images in each domain are 1670, 2048, 2344, 3929 respectively. There are seven classes: y = [dog, elephant, giraffe, guitar, horse, house, person]. We fine-tune an AlexNet-model (Krizhevsky et al., 2012), that was pre-trained on ImageNet, using ERM in combination with data augmentation. We apply SDA to select the data augmentation for the following experiment. For the domain classifier we fine-tune an AlexNet-model as described above. In addition, we use a cross-validation procedure where we leave one domain out and use the three domains for training. SDA determines four data augmentation techniques to be usefull: 'brightness', 'contrast', 'saturation', and 'hue'. In combination these four augmentations are commonly called color jitter or color perturbations. By randomly applying color perturbations we are weakening the spurious confounding influence of h_d on y, as described in Section 2. In Table 3, we compare DA to various domain generalization methods: CDANN (Li et al., 2018), L2G (Li et al., 2017b), GLCM (Wang et al., 2018), SSN (Mancini et al., 2018), IRM (Arjovsky et al., 2019), REx (Krueger et al., 2020), MetaReg (Balaji et al., 2018), JigSaw (Carlucci et al., 2019), where all methods use the same pre-trained AlexNet-model. We repeat each experiment 5 times and report the average accuracy. We find that DA obtains the highest average accuracy. The biggest performance gains of DA compared to ERM are on the test domains 'art painting' and 'sketch'. For example, the domain 'sketch' consists of black sketches of the seven object classes on white background, see Figure 7. Since the color of the object is not correlated with the class, a model relying on color features will generalize poorly to the 'sketch' domain. However, by randomly changing the colors of the images in the training domains ('art painting', 'cartoon', 'photo'), we find that DA is able to generalize much better.

Ablation study: Using all data augmentation techniques We repeat the previous experiments on Rotated MNIST, Colored MNIST, and PACS using all data augmentation techniques listed in the Appendix. We compare the accuracy of a classifier trained using all data augmentation techniques to a classifier trained using SDA. We find that using all data augmentation techniques together results in a significant drop in performance for all three datasets: 25.4% for Rotated MNIST, 8.7% for Colored MNIST, and 16.1% for PACS. We observe that there exist combinations of datasets and data augmentation techniques that lead to a drastic drop in performance on their own, e.g the PACS dataset and random rotations. We argue that a model trained without random rotations exploits the fact that, e.g, the orientation of an animal or person is usually upright. This example shows that we cannot simply describe data augmentation as 'label-preserving transformations' since a rotated animal or person will still have the same label.

5. Conclusion

In this paper, we present a causal perspective on the effectiveness of data augmentation in the context of domain generalization. By using an SCM we address a core problem of domain generalization: the spurious correlation of the domain variable d and the target variable y. While in theory, we could intervene on the domain variable d, this solution is impractical since we assume that we only have access to observational data. However, we show that data augmentation can serve as a surrogate tool for simulating interventions on the domain variable d and its children. Hereby, prior knowledge can be used to choose data augmentation techniques that only act on the non-descendants of the target variable y. Furthermore, we show that randomly applying data augmentation can be understood as randomly sampling elements from common symmetry groups. In addition, we propose a simple algorithm to select suitable augmentation techniques from a given list of transformations. We use a domain classifier to measure how well each augmentation is able to weaken the causal link between the domain d and h_d high-level features caused by d. We evaluated this approach on four different datasets and were able to show that empirical risk minimization in combination with accurately selected data augmentation results in good generalization performance. The analysis in this paper could be further used to design data augmentation to simulate interventional datasets for domain generalization methods by exploiting intervention-augmentation equivariance.