Failure Modes of Domain Generalization Algorithms

Tigran Galstyan[1,3], Hrayr Harutyunyan[2], Hrant Khachatrian[1,4], Greg Ver Steeg[2], Aram Galstyan[2]
[1] YerevaNN, [2] USC Information Sciences Institute, [3] Russian-Armenian University, [4] Yerevan State University

Abstract

Domain generalization algorithms use training data from multiple domains to learn models that generalize well to unseen domains. While recently proposed benchmarks demonstrate that most of the existing algorithms do not outperform simple baselines, the established evaluation methods fail to expose the impact of various factors that contribute to the poor performance. In this paper we propose an evaluation framework for domain generalization algorithms that allows decomposition of the error into components capturing distinct aspects of generalization. Inspired by the prevalence of algorithms based on the idea of domain-invariant representation learning, we extend the evaluation framework to capture various types of failures in achieving invariance. We show that the largest contributor to the generalization error varies across methods, datasets, regularization strengths and even training lengths. We observe two problems associated with the strategy of learning domain-invariant representations. On Colored MNIST, most domain generalization algorithms fail because they reach domain-invariance only on the training domains. On Camelyon-17, domain-invariance degrades the quality of representations on unseen domains. We hypothesize that focusing instead on tuning the classifier on top of a rich representation can be a promising direction.

1. Introduction

Over the past decade machine learning research was predominantly focused on settings where the learner observes training data from an unknown distribution and is evaluated on testing data, sampled from the same distribution. While modern deep learning approaches excel in such settings, they do significantly worse when the test data comes from a different distribution [32, 41]. These methods might rely on dataset biases to perform well, and fail when those biases are eliminated [4, 10].

The goal of generalizing beyond training distribution is formulated in the *domain generalization (DG)* task, where the learner observes training data from multiple domains and is evaluated on unseen domains. Naturally, it is assumed that training and testing domains have some invariant properties or mechanisms, which allow generalization from one to another. At a high level, all domain generalization approaches seek to capture those invariances, but do that differently. A few of the possible directions of achieving domain generalization are: learning domain-invariant representations [9, 24], learning class-conditioned domain-invariant representations [8, 21, 45], using robust loss functions [36], learning invariant causal predictors [3], and using meta-learning [20]. Most of the methods listed above outperform the straightforward empirical risk minimization (ERM) approach on toy domain generalization instances (e.g., colored MNIST). However, Gulrajani and Lopez-Paz [14] demonstrate that when evaluated on realistic DG instances, these methods are unable to outperform ERM significantly. To improve domain generalization methods or propose new ones, we need to understand why and how do domain generalization methods fail. This is the main goal of this paper.

Our contributions are threefold. First, we characterize the general failure modes of domain generalization methods: training set underfitting, test set inseparability, training-test misalignment and classifier non-invariance. We develop tools that measure the contribution of each of these failures in the total error of a given model. Inspired by the popularity of the methods based on invariant representation learning, we also characterize failure modes related to achieving domain invariance. Second, we identify two common patterns of generalization failures. In the first pattern, many algorithms achieve domain-invariant representations across the training domains, but not on unseen domains, while the generalization error is negatively correlated with the representation invariance on unseen domains. The second pattern is when domain invariance is increased across all domains, but the increase coincides with a degradation of the representations of unseen domains, thus limiting the accuracy of the models. Third, we show that by fixing the representations it is possible to isolate the classifier non-invariance failure, and significantly improve the generalization even with the most basic algorithms. These findings additionally confirm that domain-invariant representations are neither necessary nor sufficient for successful domain generalization.

2. Related Work

The ability to generalize beyond the training distribution is the key goal of machine learning. Torralba and Efros [41] show that common image classification datasets have significant differences, because of which methods trained on one dataset often fail to generalize well to other datasets. The fact that learning on a single domain is susceptible to dataset biases and spurious correlations has been confirmed in many contexts [2, 4, 6, 10, 32, 41].

A few settings focus on generalization outside the training distribution. The out-of-distribution (OOD) and robustness literature focus on the case when there is a distribution shift at test time [30, 39], including but not limited to label shifts [22, 34], co-variate shifts [12, 38], conditional shifts [44], visual distortions [7, 16], stylistic and other changes [15]. More general settings of domain generalization and domain adaptation assume that examples from multiple domains are available for training, with the difference that in domain adaptation a collection of examples (labeled or not) from the testing domain are available for adaptation [43]. In this paper we focus on the domain generalization problem (also called zero-shot domain adaptation [28]), because of its generality and better correspondence with practical settings. Nevertheless, most of the proposed techniques and definitions can be easily extended to domain adaptation.

A group of approaches aim to get domain generalization by learning domain-invariant representations [8, 9, 21, 24, 45]. DANN [9] uses an adversarial classifier to predict domain from representations, while C-DANN [21] learns such a classifier for each domain separately. Galstyan et al. [8] regularize empirical risk minimization (ERM) with a term that uses Hilbert-Schmidt independence criterion (HSIC) [11] to make representations be independent from domains conditioned on labels. Zhao et al. [45] use a variety of techniques to enforce the distribution of labels conditioned on representations be the same for all domains. DeepCORAL [40] adds a regularization term to align the second-order statistics of representations of different domains. Invariant risk minimization (IRM) [3] aims to learn invariant causal predictors, by finding a representation of data such that optimal classifiers on top of representations are the same for each domain. Li et al. [21] use meta-learning with the one-step look-ahead gradient update technique to simulate evaluating on unseen domains during the training. GroupDRO [36] minimizes the worst-case risk across training domains. Recently, methods based on gradient matching have been proposed [31, 37]. Finally, a few works introduce benchmarks for evaluating domain generalization methods [14, 18].

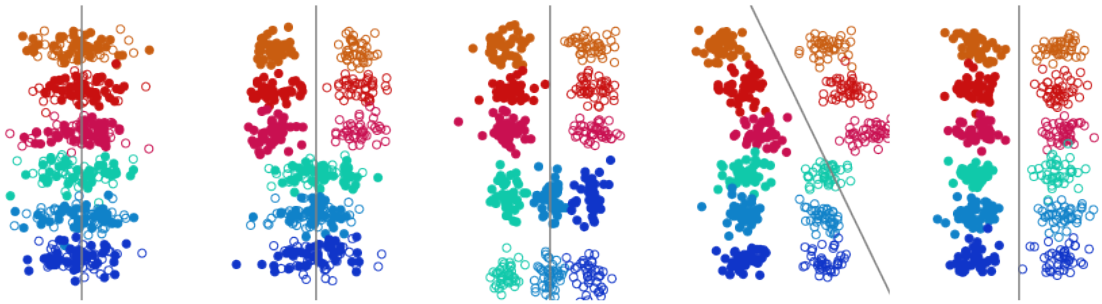
Gulrajani and Lopez-Paz [14] demonstrate that none of the existing domain generalization approaches outperform empirical risk minimization when evaluated on realistic tasks. There is a very limited amount of research done on why and how these domain generalization methods fail. Rosenfeld et al. [33] study the failure modes of the IRM in theoretical settings. Nagarajan et al. [25] explain the mechanisms by which ERM fails on very easy out-of-domain generalization tasks. In contrast to these works, our analysis and proposed techniques below are applicable to any domain generalization algorithm.

3. Problem Setting and Notation

Consider an input space $\mathcal{X}$ and output space $\mathcal{Y}$. A joint probability distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ is called a domain and defines a prediction task. In the domain generalization task we assume there is a family $\mathscr{D}$ of domains that are somehow related to each other and correspond to similar prediction tasks. The learner observes training data from n_1 domains, $p^1_1(x, y), \ldots, p^1_{n_1}(x, y)$. The goal is to learn a predictor $f : \mathcal{X} \to \mathcal{Y}$ that generalizes to unseen domains from $\mathscr{D}$. Note that in contrast to the domain *adaptation* problem, here the learner cannot do any adaptation at inference time.

In this paper we focus on classification tasks, where $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \ldots, C\}$. We assume that, in addition to n_1 training domains, we have n_2 validation domains $p^2_1(x, y), \ldots, p^2_{n_2}(x, y)$, and n_3 test domains, $p^3_1(x, y), \ldots, p^3_{n_3}(x, y)$. We assume that there is no label shift across domains: $p^1_1(y) = \ldots = p^1_{n_1}(y) = p^2_1(y) = \ldots = p^2_{n_2}(y) = p^3_1(y) = \ldots = p^3_{n_3}(y)$. We discuss this limitation in Appendix F. We also assume that for each domain $i = 1, \ldots, n_j$, $j = 1, 2, 3$, we are given a collection of independent samples $\mathcal{D}^j_i$ from the corresponding distribution $p^j_i(x, y)$. Each of these sets is further divided into two parts: $\mathcal{D}^j_i = \mathcal{T}^j_i \cup \mathcal{V}^j_i$. For simplicity, we also define the union of the samples of training (validation, testing) domains: $\mathcal{D}^j = \bigcup_{i=1}^{n_j} \mathcal{D}^j_i$, $\mathcal{T}^j = \bigcup_{i=1}^{n_j} \mathcal{T}^j_i$, $\mathcal{V}^j = \bigcup_{i=1}^{n_j} \mathcal{V}^j_i$, for each $j = 1, 2, 3$. The algorithms will be trained on samples from $\mathcal{T}^1$. The set $\mathcal{V}^1$ is called *in-domain validation set* in [18], *quasi-development set* in [8] and *training-domain validation set* in [14]. It is used to measure the performance of the algorithms on unseen samples from the training domains. The performance on unseen domains is measured using $\mathcal{D}^3$. The sets $\mathcal{T}^j$ and $\mathcal{V}^j$, $j = 2, 3$, are used for analysis.

We consider domain generalization methods that use neural networks with two components: a feature extractor $z = h_\theta(x) \in \mathbb{R}^d$ with parameters $\theta \in \Theta$, and a classification head $\widehat{y} = f_w(z) \in \mathbb{R}^C$ with parameters $w \in \mathcal{W}$. Throughout the paper we call z the *representation* of x. Let $\ell : \mathbb{R}^C \times \mathcal{Y} \to \mathbb{R}$ be a loss function that measures discrepancy between a prediction and a label. In this work ℓ is chosen to be the standard 0-1 loss function: $\ell(\widehat{y}, y) = \mathbf{1}\{\text{argmax}_k \widehat{y}_k(z(x)) \neq y\}$, which is the most popular evaluation metric in classification tasks. Nevertheless, most of the results of this paper can be easily extended to other choices of loss functions, such as the negative log-likelihood loss function, often used for *training* classifiers.



(a) $e'_0 \gg 0$ (b) $e'_0 = 0, e_1 \gg 0$ (c) $e'_1 = 0, e_2 \gg 0$ (d) $e'_2 = 0, e_3 \gg 0$ (e) $e'_3 = 0$

Figure 1. Failure modes of domain generalization algorithms demonstrated by 2D representations of a two-class data. Failure modes are described in Sec. 4. Empty and full circles correspond to 0 and 1 classes. Colors encode the domains, training domains use colors close to red, test domains use colors close to blue. The lines correspond to decision boundaries likely to be found by a simple ERM algorithm trained on the training domains. In all subfigures the domains are distinguishable: $d_0 \gg 0$.

4. Failure Modes

We propose simple evaluation metrics to diagnose a trained model and identify a set of failures that contribute to the final error on unseen domains. We present simplified schematic visualizations of 2-dimensional representation spaces corresponding to each of the failure modes, *e.g.* Fig. 1. In all such figures each circle corresponds to a representation of one sample. The domain of a sample is encoded by the color of its circle. Orange-red-pink colors correspond to the training domains, while green-blue colors correspond to the test domains. Filled and empty circles are used to encode the binary labels.

To formally define the generalization and invariance metrics, we need the following additional notation. Let $(X^1_1, Y^1_1), \dots (X^1_{n_1}, Y^1_{n_1})$ be random variables drawn from training domains $p^1_1(x, y), \dots, p^1_{n_1}(x, y)$ and $(X^3_1, Y^3_1), \dots (X^3_{n_3}, Y^3_{n_3})$ be random variables drawn from test domains $p^3_1(x, y), \dots, p^3_{n_3}(x, y)$. Let $(X^{1,3}, Y^{1,3})$ be a random variable drawn from the mixture of all training and test domains $p^{1,3}(x, y) = \frac{1}{n_1+n_3}\left(\sum_{i=1}^{n_1} p^1_i(x, y) + \sum_{i=1}^{n_3} p^3_i(x, y)\right)$.

4.1. Generalization metrics

Below we define four evaluation metrics that capture qualitatively different aspects of generalization. All of this metrics will be some kind of errors (so lower better). Hence, we will use terms "metric" and "error" interchangeably.

Training set underfitting. How well does the model perform on training domains? Formally, this metric is denoted by e'_0 and is defined as follows:

$$e'_0 \triangleq \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbb{E}_{X^1_i, Y^1_i}\left[\ell\left(f_w(h_\theta(X^1_i)), Y^1_i\right)\right],$$

where h_θ and f_w are the learned feature extractor and classifier, respectively. This metric is expected to be small for most domain generalization algorithms. A model might have large e'_0 if the regularization terms of its objective were so strong compared to the classifier loss that the feature extractor failed to learn anything useful. An example will be discussed in the Experiments section. Training set underfitting is demonstrated in Fig. 1a.

Test set inseparability. How well are the representations of the test domains separable with respect to the chosen class of classifier heads? Formally, we define test set inseparability error as follows:

$$e'_1 \triangleq \inf_{w' \in \mathcal{W}} \left\{ \frac{1}{n_3}\sum_{i=1}^{n_3} \mathbb{E}_{X^3_i, Y^3_i}\left[\ell\left(f_{w'}(h_\theta(X^3_i)), Y^3_i\right)\right] \right\}.$$

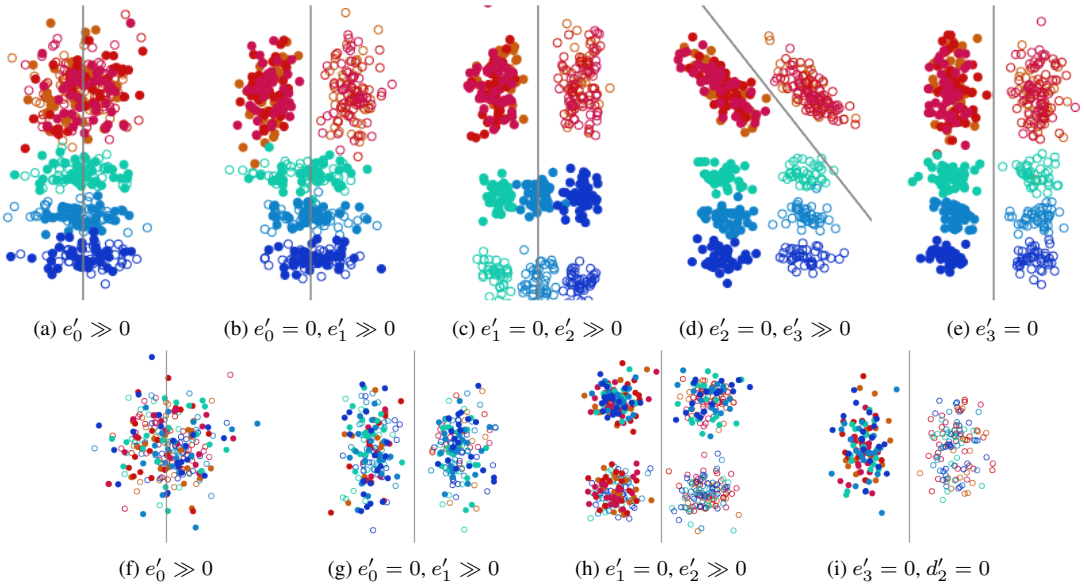
This metric will be large for models whose feature extractor is overfitted on the training domains and does not produce a reasonable representation for the test domains. Note that the quality of the representation is measured with respect to the class of classifier heads (e.g., linear functions), because we are not interested in cases when the representations have information about domains but that information is not decodable/usable by the considered family of classifiers. The case when $e'_0 = 0$ and e'_1 is large is demonstrated in Fig. 1b.

Training-test misalignment. Is there a common classifier for the representations of training and test domains? The extent of the answer to this question being negative is measured by the e'_2 metric:

$$e'_2 \triangleq \frac{1}{n_3}\sum_{i=1}^{n_3} \mathbb{E}_{X^3_i, Y^3_i}\left[\ell(f_{\tilde{w}}(h_\theta(X^3_i)), Y^3_i)\right],$$

$$\text{where } \tilde{w} \in \arg\min_{\tilde{w} \in \mathcal{W}} \mathbb{E}_{X^{1,3}, Y^{1,3}}\left[\ell(f_{\tilde{w}}(h_\theta(X^{1,3})), Y^{1,3})\right].$$

If e'_1 is small, the representations of the samples from each domain are good enough to be separated by a domain-specific classifier. But it might be impossible to find a classifier that would separate samples from all domains (both



(a) $e'_0 \gg 0$ (b) $e'_0 = 0, e'_1 \gg 0$ (c) $e'_1 = 0, e'_2 \gg 0$ (d) $e'_2 = 0, e'_3 \gg 0$ (e) $e'_3 = 0$

(f) $e'_0 \gg 0$ (g) $e'_0 = 0, e'_1 \gg 0$ (h) $e'_1 = 0, e'_2 \gg 0$ (i) $e'_3 = 0, d'_2 = 0$

Figure 2. Failure modes of domain generalization algorithms demonstrated by 2D representations of a two-class data. The first row corresponds to training domain invariance ($d'_0 = 0$), but $d'_1 \gg 0$. The second row corresponds to training-test domain invariance ($d'_1 = 0$). In particular, the first three images demonstrate that $d'_1 = 0$ can co-occur with large e_0, e_1 and e_2. The right-most image demonstrates that $e'_0 = 0$ and $d'_2 = 0$ together imply $e'_3 = 0$. Failure modes are described in Sec. 4.

training and test) at once, resulting in relatively high e'_2 error. This case is demonstrated in Fig. 1c. Such scenarios can be thought of as "milder" versions of overfitting, as the feature extractor learned useful and decodable information, but was not able to distribute the representations in a consistent way across domains.

Classifier non-invariance. This final generalization metric is the standard test error and measures the performance of the learned model on test domains. Formally, the metric e'_3 is defined as follows:

$$e'_3 \triangleq \frac{1}{n_3} \sum_{i=1}^{n_3} \mathbb{E}_{X_i^3, Y_i^3} \left[\ell(f_w(h_\theta(X_i^3)), Y_i^3) \right].$$

If $e'_2 = 0$, the representations are so good that there exists a classifier that can separate samples from both training and test domains with significant success. In such cases e'_3 essentially measures whether the training algorithm was able to find a classifier that works for both training and test domains. Hence, the name of this metric e'_3 is "classifier non-invariance". Note that a training algorithm might easily fail to select an invariant classifier, as it does not have access to the test data during training. This scenario is schematically demonstrated in Fig. 1d.

Note that the four failure modes defined above and depicted in Fig. 1 are extreme cases and should not be thought as a comprehensive list of cases. In Sec. 4.4 we propose a way of attributing the model's overall test error to each of these failure modes.

4.2. Invariance metrics

Many domain generalization algorithms attempt to learn domain-invariant representations. The intuition is that if the representations of the samples are similar across all domains, then a classifier designed for the training domains will generalize to the test domains. As the algorithms have access only to the training data, they usually achieve domain invariance across the training domains, but not across the union of training and test domains.

Motivated by the generalization metrics defined above, we introduce metrics that assess qualitatively different aspects of domain invariance of learned representation. While there are many ways to measure the extent of invariance of representations across two or more domains (e.g., using formal distances or divergences between probability distributions), we choose to use a technique similar to the one used in generalization metrics above. Informally, we will measure how invariant are the representations of samples of two domains by measuring how well one can differentiate them using a domain classifier. Importantly, we will draw domain classifier from the same family of functions as label classifiers. That is, if the label classifier head uses a specific architecture, we would use the same architecture (with a different number of outputs) for domain classifiers. This intentionally ignores domain information in representations that cannot be decoded by the label classifier. Domain classifiers will be functions of the form $g_\omega : \mathbb{R}^d \to \mathbb{R}^K, \omega \in \Omega$, where K is the number of domains.

Training domain distinguishability. Are the representations of examples from the training domains domain-invariant? Formally, we denote our first invariance metric d'_0 and define it the following way:

$$d'_0 \triangleq 1 - \inf_{\omega \in \Omega} \left[\frac{1}{n_1} \sum_{i=1}^{n_1} \mathbb{E}_{X_i^1} \left[\ell(g_\omega(h_\theta(X_i^1)), i) \right] \right] - \frac{1}{n_1}.$$

Here the constant $\frac{1}{n_1}$ is the accuracy of the random classifier. Note that lower values of d'_0 correspond to higher invariance, and $d'_0 = 0$ implies it is impossible to distinguish domains better than the trivial baseline. Large values of d'_0 can arise when the regularizer designed to induce invariance is too weak compared to the classification loss. In practice, most algorithms do not end up in this state if the regularization term is tuned correctly. It is important to note that achieving domain invariance on the training domains is not necessary to have domain generalization, as demonstrated in Fig. 1e.

Training-test domain distinguishability. Are the representations of examples from the union of the training and test domains domain-invariant? Our second invariance metric d'_1 measures the extent of this question having a positive answer:

$$d'_1 \triangleq 1 - \inf_{\omega \in \Omega} \mathbb{E}_{X^1_{1:n_1}, X^3_{1:n_3}} \left[\frac{1}{n_1 + n_3} \left(\sum_{i=1}^{n_1} \ell(g_\omega(h_\theta(X_i^1)), i) + \sum_{i=1}^{n_3} \ell(g_\omega(h_\theta(X_i^3)), i + n_1) \right) \right] - \frac{1}{n_1 + n_3}.$$

Assuming that $d_0 = 0'$, this metric d'_1 can be large if the distributions of the representations of the training and test sets do not coincide; if there is no domain invariance across the test domains; or both. In theory, this failure of reaching invariance can coincide with all failure modes of generalization, as shown in Fig. 2. Moreover, it is possible to have domain generalization along with large d'_1 (Fig. 2e).

Even when a model achieves high training-test domain invariance (low d'_1), it is still possible that representations of Class 1 samples of the first domain coincide with the representations of Class 2 samples of the second domain and vice versa. In this case, there will be no domain generalization. The next metric is designed to capture such situations.

Training-test class-conditional domain invariance. Are the representations of samples belonging to each of the classes domain-invariant? Let E_y denote the event $(Y_1^1 = y \wedge \cdots \wedge Y_{n_1}^1 = y \wedge Y_1^3 = y \wedge \cdots \wedge Y_{n_3}^3 = y)$. We define the d'_2 metric as follows:

$$d'_2 \triangleq 1 - \frac{1}{C} \sum_{y=1}^{C} \inf_{\omega \in \Omega} \mathbb{E} \left[\frac{1}{n_1 + n_3} \left(\sum_{i=1}^{n_1} \ell(g_\omega(h_\theta(X_i^1)), i) + \sum_{i=1}^{n_3} \ell(g_\omega(h_\theta(X_i^3)), i + n_1) \right) \middle| E_y \right] - \frac{1}{n_1 + n_3}.$$

This d'_2 metric can be seen as the conditional version of the previous metric d'_1.

4.3. Relations between domain invariance and generalization metrics

We prove two propositions that establish connections between generalization and invariance failures. In particular, these propositions rule out some combinations of invariance and generalization failures. We first formally define domain invariance of representations and class-conditional domain invariance of representations (not to be confused with the corresponding invariance metrics).

Definition 4.1 (Domain invariance of representations)**.** *Let $\mathscr{D}$ be a family of domains. We say that the representations learned by a feature extractor $z = h(x)$ are domain-invariant with respect to $\mathscr{D}$ if for any two domains $p_1(x, y)$ and $p_2(x, y)$ from $\mathscr{D}$, the distributions of representations are equal, i.e., $\forall z, p_{Z_1}(z) = p_{Z_2}(z)$, where $X_1 \sim p_1(x), X_2 \sim p_2(x), Z_1 = h(X_1)$ and $Z_2 = h(X_2)$.*

Likewise, we define invariance of representations conditioned on labels by requiring $p(z|y)$ to be the same for across all domains of the family $\mathscr{D}$.

Proposition 1. *If $e'_2 = 0$, then domain invariance of representations w.r.t. the union of training and testing domains implies class-conditioned domain invariance w.r.t. the union of training and testing domains.*

Proposition 2. *If $e'_0 = 0$ and representations are class-conditioned domain-invariant w.r.t. the union of training and testing domains, then $e'_3 = 0$.*

The second proposition implies that for a class-conditioned domain-invariant model, perfect performance on the training domains is sufficient for domain generalization.

4.4. Decomposition of errors

The generalization metrics defined above are sequential in the sense that if e'_i is large then e'_{i+1} is likely to be large as well. For this reason, the differences $e'_{i+1} - e'_i,\ i = 0, 1, 2$, are more suitable quantities for analysis purposes. In fact, the failure modes depicted in Fig. 1 are cases when one of these differences is large in conjunction with the previous metric being close to zero. Following this reasoning, we define $e_0 \triangleq e'_0$, $e_1 \triangleq e'_1 - e'_0$, $e_2 \triangleq e'_2 - e'_1$, $e_3 \triangleq e'_3 - e'_2$, and decompose the error $e \triangleq e'_3$ on the test domains into four components:

$$e = \underbrace{e_0}_{\text{training set underfitting}} + \underbrace{e_1}_{\text{test set inseparability}} + \underbrace{e_2}_{\text{training-test misalignment}} + \underbrace{e_3}_{\text{classifier non-invariance}}. \quad (1)$$

Each of these components can be interpreted as the individual contribution of the corresponding failure mode to the error of the model. This decomposition is the most meaningful when its components are nonnegative. In general, e_1, e_2, and e_3 can be negative, for example when samples of testing domains are significantly easier to classify compared to those of training domains. However, this is a rare phenomenon and has not been observed in our experiments. Moreover, one can prove that on average (w.r.t. to the training-test splits of the domains) each $e_i,\ i = 0, 1, 2, 3$ is nonnegative. We give the precise formulation of this statement along with its proof in Appendix B.

Similarly, domain-distinguishability $d \triangleq d'_2$ can be decomposed into three components:

$$d = \underbrace{d_0}_{\text{training domain distinguishability}} + \underbrace{d_1}_{\text{training-test domain distinguishability}} + \underbrace{d_2}_{\text{training-test class-conditional domain distinguishability}}, \quad (2)$$

where $d_0 \triangleq d'_0$, $d_1 \triangleq d'_1 - d'_0$, and $d_2 \triangleq d'_2 - d'_1$. Again, each of these components can be interpreted as the individual contribution of the corresponding failure mode. Some components of this decomposition also can be negative in rare situations but are nonnegative if we consider averaging over training-test splits of the domains (see Appendix B).

5. Experiments

5.1. Datasets and algorithms

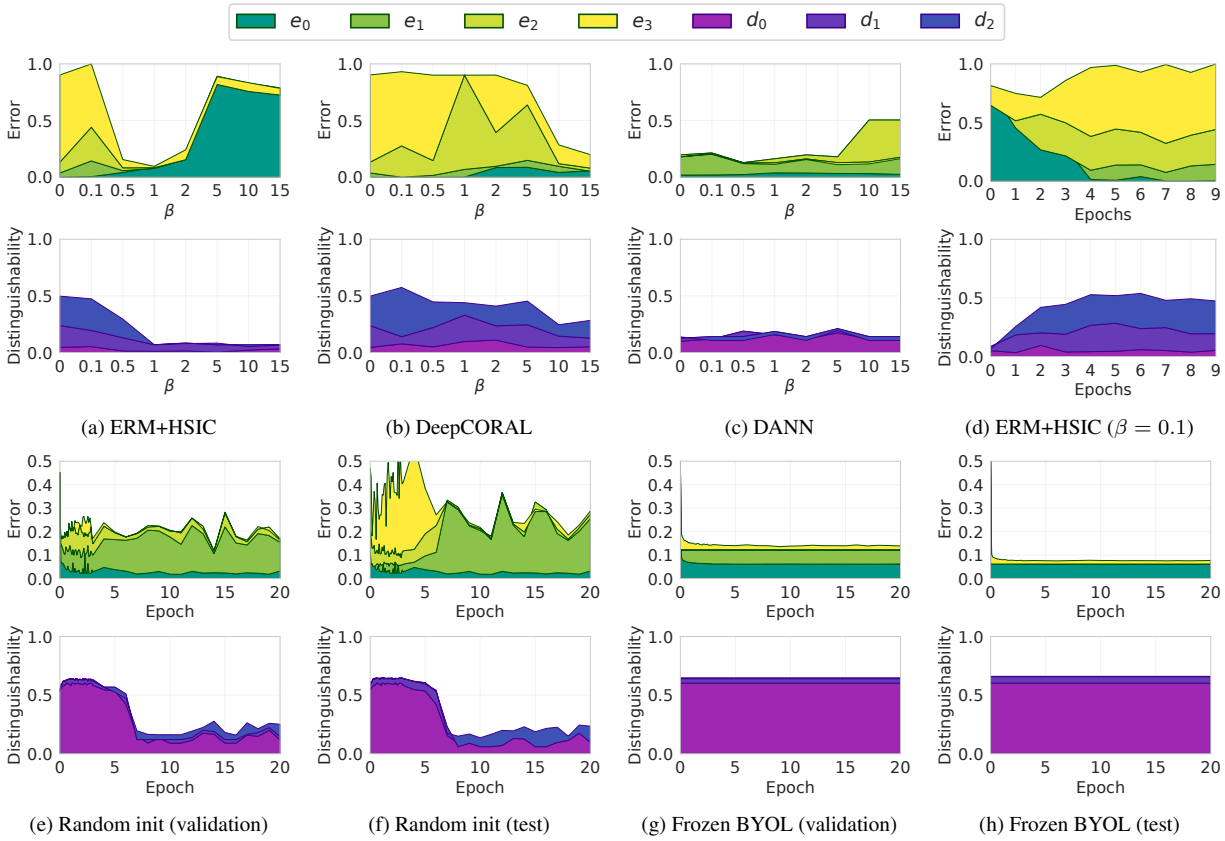
The number of datasets suitable for testing domain generalization algorithms is not large. An attempt to collect them under a unified format was done in [14]. Two of the seven datasets are based on MNIST digits, four others are simple unions of unrelated datasets with similar labels, and only one is realistic: TerraIncognita. This latter one has a serious label shift between domains. Recently proposed WILDS benchmark [18] has another seven datasets which are connected to real-world problems. Only one of the seven, **Camelyon17**, is carefully designed to have no label shift. It is a patch-based variant of the larger Camelyon17 dataset [5] of lymph node captures. The version in the WILDS benchmark contains 450000 96x96 patches of images of cancer metastases in lymph node sections. The label for each patch is binary indicating whether the patch contains any tumor tissue. The domain of a patch is the hospital from where the image comes from. There are five different hospitals, three for training, one for validation and one for testing. Following [18], we use Densenet-121 [17] for all algorithms on this dataset and train for 10 epochs.

We also use a slightly modified version of **Colored MNIST** dataset from [8], originally from [19]. We changed the dataset generation process to mimic Camelyon17. Namely, we construct five domains by splitting MNIST images into five sets of equal size. For each domain we randomly fix three "colors", which are essentially 50 dimensional vectors, one for each digit. Then we "colorize" each image with the corresponding color. We end up with around 1200 images in each domain of shape 50x28x28. All models trained on this dataset use a simple neural network with a two-layer ReLU-activated convolutional feature extractor and a linear layer on top of it. All models are trained for 10 epochs. Samples from the datasets are presented in Appendix C.

We analyze the following domain generalization algorithms. Empirical Risk Minimization (**ERM**) [42] is used as a baseline. It minimizes the sum of errors on all training domains and does not use domain information. **ERM+HSIC** [8] is a simple algorithm that attempts to induces domain invariance for training domains by adding a Hilbert-Schmidt Independence Criterion (HSIC) [12] regularization term that penalizes domain information in the learned representations. **DeepCORAL** [40] was introduced as a domain adaptation algorithm, but was recently applied in domain generalization settings [14]. It penalizes the difference between the means and covariance matrices of representation distributions across domains. Invariant Risk Minimization (**IRM**) algorithm [3] tries to push representation distributions for all domains to have the same optimal classifier head. In [29] the authors analyze the phenomenon of gradient starvation in algorithms based on gradient descent. They propose a new regularization method called Spectral Decoupling (**SD**). To overcome gradient starvation, it penalizes model's confidence by adding an L2 penalty on networks logits. **GroupDRO** was used by [35] to tackle poor worst-group performance. It tries to increase worst-group performance by avoiding spurious correlations in the training data. By interpreting domains as groups, this algorithm becomes applicable to our setting. Another method relying on domain-invariant features is Domain-Adversarial Neural Network (**DANN**) introduced in [1]. It tries to achieve invariance by using a domain classifier (from features) and a gradient reversal layer. All of these algorithms (except ERM) have a regularization strength hyperparameter, which we denote by β. Hyperparameter ranges can be found in Appendix D.

Following literature, we consider the outputs of the last hidden layer as the learned representations $h_\theta(x)$. Hence, label and domain classifiers $f_w(z)$ are linear (i.e., just a single fully-connected layer).

Model selection. The authors of DomainBed benchmark explicitly warned against improper model selection methods in domain generalization. They advocated that in practice model selection should be done either based on the performance on in-domain validation sets ($\mathcal{V}^1$), or by using leave-one-domain-out validation. WILDS benchmark introduced separate validation domains and suggested to use them for model selection. In our experiments we will use this method



(a) ERM+HSIC (b) DeepCORAL (c) DANN (d) ERM+HSIC ($\beta = 0.1$)

(e) Random init (validation) (f) Random init (test) (g) Frozen BYOL (validation) (h) Frozen BYOL (test)

Figure 3. Decomposition of generalization errors and domain-distinguishability of several models measured on the validation domains (a-e, g) and test domains (f, h). Horizontal axis corresponds to: (a)-(c) regularization strength of the algorithms, (d-h) training epochs. Figure (c), (b) and (d) show results on Colored MNIST dataset; all others are on Camelyon17.

for hyperparameter selection. To select the best checkpoint inside a single run, one can use the same validation domains. In this paper we always picked the last epoch.

5.2. Measuring the failures

To avoid accessing the test domains, we perform analysis on training and validation domains only. The metrics defined in Sec. 4 contain expectations over distributions and infimums over classifiers. In practice, we approximate infimums by empirical risk minimization using the implementation of logistic regression in scikit-learn package [27]. Note that this might fail to find the optimum, and the empirical estimates might be worse than the true values. In case of e'_2, the estimate can be worse than the solution found by the domain generalization algorithm itself, i.e. $e'_3 < e'_2$. In fact, we encounter this phenomenon later in Tab. 1. Following standard machine learning practices, we approximate distributions $p_i^1(x, y)$ and $p_j^2(x, y)$ with corresponding empirical distributions $\mathcal{T}_i^1$ and $\mathcal{T}_j^2$ during training, and with $\mathcal{V}_i^1$ and $\mathcal{V}_j^2$ when reporting the scores ($i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$). In practice, to make d'_1 and d'_2 comparable with d'_0, we train and evaluate domain classifiers on the union of $n_1 - n_2$ training domains and n_2 validation domains.

6. Results and Discussion

Our analysis shows distinct patterns of failures on the two datasets. On **Colored MNIST**, all algorithms, including the ERM baseline ($\beta = 0$), achieve domain invariance on training domains, but unseen domains remain quite distinguishable. At the same time, the representations of each domain remain separable (e_0 and e_1 are close to zero), but we have well-expressed training-test misalignment and/or classifier non-invariance across all algorithms and hyperparameters (Figs. 3b, 6b and 6c). The error on the validation set is usually larger than 0.8. We observe a positive correlation (0.4-0.5) between d'_1 and generalization error on the validation set (e'_3) (Fig. 5a).

The only exception to this pattern is seen on ERM+HSIC algorithm (Fig. 3a). When the regularization strength β approaches 1, we achieve full invariance across all domains, training-test misalignment approaches zero and the classifier becomes invariant. The error on the test set is 0.09 which is

Algorithm	Initialization	Valid.	Test
ERM (Fig. 3e)	Random	0.164	0.287
ERM	Frozen BYOL	0.138	0.077
ERM (wd=0, Fig. 3g)	Frozen BYOL	0.097	0.070
SD ($\beta = 5$)	Frozen BYOL	0.111	0.070
GroupDRO ($\beta = 5$)	Frozen BYOL	0.140	0.078
IRM ($\beta = 5$)	Frozen BYOL	0.147	0.081
e'_2 (lower bound, est.)	Frozen BYOL	0.123	0.062

Table 1. Generalization error e'_3 on validation and test domains of several algorithms trained on Camelyon17. Weight decay coefficient for all algorithms is set to 0.01 except the third row. The last row shows the generalization error without classifier non-invariance component (e'_2). As the representations are fixed, these values remain constant during the training and serve as a lower bound for all models with frozen representations. See Sec. 5.2 on why the estimated lower bound is poor on the validation set.

equal to the error on the training set. Stronger regularization makes the error on the training set larger as the representations collapse. The latter causes the correlation between d'_1 and e'_3 to be much lower for ERM+HSIC (Fig. 5b).

Another interesting phenomenon was observed while training ERM+HSIC with a lower $\beta = 0.1$. Initially, the model does not fit the training domains (large e_0). Over time, e_0 is gradually transformed into e_3, e_2 and e_1 (Fig. 3d). The overall error always stays above 0.7 during the training, but the composition of the error changes significantly.

On **Camelyon17**, training domain distinguishability and training-validation domain distinguishability are virtually the same for all algorithms (Figs. 3c and 7). Nevertheless, validation set inseparability is always non-zero, which is the largest contributor to the error on unseen domains for well performing models. The magnitude of e_1 varies a lot during the training and across random seeds (Appendix E). Additionally, there is no correlation between d'_1 and e'_3 (Figs. 5c and 5d). This gives a little hope that focusing on domain invariance will help succeed on this dataset.

A more detailed analysis uncovers an interesting pattern. As we have only two unseen domains in Camelyon17, we look at both of them during our analysis (violating "do not look at the test set" rule). Many methods we tried, including the baseline, increase the invariance after the first few epochs of training. The large variance of e_1 is observed only *after* d_1 gets small (Figs. 3e and 3f). Instead, when the domains are well distinguishable, e_1 is relatively stable (although the absolute values are different on the two domains) and is accompanied by e_2 and e_3. In fact, on the test domain, the largest contributor to the generalization error over the first epochs is e_3. This implies that if we do not push the representations to be domain-invariant, there is a hope to get better generalization on unseen domains by focusing on the classifier. Unfortunately, even the basic ERM, without additional loss terms, converges towards training-domain invariance and harms test set separability.

To verify that focusing on the classifier can be beneficial, we take a ResNet-50 pretrained on ImageNet using BYOL algorithm [13] and fix the representations. At this point, the representations of different domains are well distinguishable (similar to the first epochs of a regular training), e'_1 for the validation and test domains are 0.123 and 0.062 respectively. This implies that the test domain is more similar to the training domains than the validation domain in BYOL's space. As we freeze the representations, these numbers serve as lower bounds for the generalization error, and the training can affect only e_0 and e_3. Experiments show that the ERM baseline with no weight decay (Figs. 3g and 3h), as well as SD algorithm with a large β, can indeed decrease e_3 and approach the lower bound (Tab. 1). GroupDRO and IRM fail to decrease e_3, while DANN and ERM+HSIC are not applicable to the setup with fixed representations. This result indicates that avoiding even a little collapse of representations, which in these cases coincides with increased domain invariance, opens new opportunities to decrease the generalization error by targeting just e_3. The causal link between poor representations and domain invariance is not clear.

Model selection. To verify whether model selection using validation domains is suitable, we compute the correlation between accuracy of the models on validation and test domains at each epoch for a given dataset/algorithm pair. On Colored MNIST we get 0.98, 0.8, 0.73 and 0.31 correlations for ERM+HSIC, DeepCORAL, SD and DANN, respectively. On Camelyon17 basically no correlation is observed, which is similar to the findings in [23]. Corresponding plots are shown in Appendix H.

Model complexity. To test the dependence of generalization failures on model complexity, we followed [26] to train several ResNet 18K models ($k = 2, 4, 8, 16, 32, 64$) with ERM+HSIC objective on Colored MNIST ($\beta = 1$). As seen on Fig. 6d, larger models better fit the training set, but get worse training-test domain invariance (d'_1).

7. Acknowledgements

This work is based in part on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation therein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Laboratory, DARPA or the U.S. Government. TG and HK were supported by the RA Science Committee, in the frames of the research project No. 20TTAT-AIa024. HH was supported by a USC Annenberg Fellowship.

References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *stat*, 1050:15, 2014. 6

[2] Ehab A AlBadawy, Ashirbani Saha, and Maciej A Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, 45(3):1150–1158, 2018. 2

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 2, 6

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 456–473, 2018. 1, 2

[5] P. Bándi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. Ehteshami Bejnordi, B. Lee, K. Paeng, A. Zhong, Q. Li, F. G. Zanjani, S. Zinger, K. Fukuta, D. Komura, V. Ovtcharov, S. Cheng, S. Zeng, J. Thagaard, A. B. Dahl, H. Lin, H. Chen, L. Jacobsson, M. Hedlund, M. Çetin, E. Halıcı, H. Jackson, R. Chen, F. Both, J. Franke, H. Küsters-Vandevelde, W. Vreuls, P. Bult, B. van Ginneken, J. van der Laak, and G. Litjens. From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019. 6

[6] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE, 2018. 2

[7] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017. 2

[8] Tigran Galstyan, Hrant Khachatrian, Greg Ver Steeg, and Aram Galstyan. Robust classification under class-dependent domain shift. *arXiv preprint arXiv:2007.05335*, 2020. 1, 2, 6

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 2

[10] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1, 2

[11] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005. 2

[12] A. Gretton, A.J. Smola, J. Huang, Marcel Schmittfull, K.M. Borgwardt, Bernhard Schölkopf, J. Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning, 131-160 (2009)*, 01 2009. 2, 6

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 8

[14] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. 1, 2, 6

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 2

[16] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 2

[17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[18] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020. 2, 6

[19] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 6

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1

[21] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 1, 2

[22] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018. 2, 14

[23] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021. 8

[24] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013. 1, 2

[25] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2021. 2

[26] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. 8

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 7

[28] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 764–781, 2018. 2

[29] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint arXiv:2011.09468*, 2020. 6

[30] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009. 2

[31] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. *arXiv preprint arXiv:2109.02934*, 2021. 2

[32] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019. 1, 2

[33] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. 2

[34] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002. 2

[35] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019. 6

[36] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. 1, 2

[37] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021. 2

[38] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 2

[39] Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30:3–28, 2009. 2

[40] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 2, 6

[41] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 1, 2

[42] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 6

[43] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[44] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013. 2

[45] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2