A RATE-DISTORTION APPROACH TO DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization deals with the difference in the distribution between the training and testing datasets, i.e., the domain shift problem, by extracting domain-invariant features. In this paper, we propose an information-theoretic approach for domain generalization. We first establish the domain transformation model, mapping a domain-free latent image into a domain. Then, we cast the domain generalization as a rate-distortion problem, and use the information bottleneck penalty to measure how well the domain-free latent image is reconstructed from a compressed representation of a domain-specific image compared to its direct prediction from the domain-specific image itself. We prove that the information bottleneck penalty guarantees that domain-invariant features can be learned. Lastly, we draw links of our proposed method with self-supervised contrastive learning without negative data pairs. Our empirical study on two different tasks verifies the improvement over recent baselines.

1 INTRODUCTION

Deep neural networks (DNNs) are highly expressive models that reach state-of-the-art performance in challenging tasks, such as speech and visual recognition recognition (Devlin et al., 2018; He et al., 2016). DNNs can achieve such performance by capturing complex correlations among input elements, e.g., pixels of an image. However, such correlations might also contain spurious features (Szegedy et al., 2013; Beery et al., 2018; Alcorn et al., 2019) that hurt the generalization performance of DNNs on out-of-distribution tasks. Unfortunately, real-world applications often encounter such out-of-distribution samples, e.g., when the training domain does not match the testing domain. A prominent example is deblurring, where models are trained on simulated blurs which differ substantially to real-world blurring (Koh et al., 2021). In other words, generalization across domains is a critical task before deploying DNNs to real-world application.

The literature on domain generalization aims to learn the features that are invariant across multiple training domains, and to use those features for out-of-distribution generalization. In domain generalization, multiple source domains are accessible during training, but the target domains are not (Blanchard et al., 2011; Muandet et al., 2013). Invariant risk minimization (IRM) (Arjovsky et al., 2019) is a prominent approach for learning domain invariant features. However, IRM suffers from the case when the invariant features contains full information about the label (Ahuja et al., 2020). To overcome this shortcoming, Ahuja et al. (2021) introduce the information bottleneck theory on neural networks (Tishby & Zaslavsky, 2015), and show that their method will be guaranteed to converge to the invariant features. Empirical works on the domain generalization align source domain distributions for domain-invariant representation learning (Sun & Saenko, 2016; Li et al., 2018b;c), meta learning (Li et al., 2018a; Balaji et al., 2018), augmenting data with image information (Zhou et al., 2020), or constructing the contrastive batch (Kim et al., 2021). Despite their success, there is no guarantee that empirical methods can solve the task across different environments.

In this paper, we use an information-theoretic approach to tackle domain generalization. We assume there is a domain-free latent instance (e.g., an image) that captures the invariant features we want to extract. We define a domain transformation model that maps the domain-free latent instance into a domain and then we apply the rate distortion theory to obtain a domain-invariant representation. The proposed method, called Twins, is guaranteed to converge to the invariant feature under the linear classification structural equation model (Ahuja et al., 2021). We evaluate our method on linear unit

tests (Aubin et al., 2021) and variants of MNIST dataset (LeCun & Cortes, 2010; Xiao et al., 2017; Clanuwat et al., 2018), which validates the theoretical analysis and demonstrates how the proposed method can outperform the previous ones. Our contributions can be summarized as follows:

- We cast domain generalization as a rate distortion problem and prove how the proposed method can converge.
- We illustrate how the proposed method extends previous results on domain generalization, and draw links to self-supervised contrastive learning.
- We demystify the success of contrastive learning by giving a contrastive learning based domain generalization algorithm with theoretical guarantee.
- We evaluate our method on two datasets and observe consistent improvement over existing baselines.

2 PRELIMINARY ON DOMAIN GENERALIZATION

Assume that the instance-label pair (X, Y) is distributed according to an unknown distribution $\mathbb{P}(X, Y)$. The objective of standard supervised learning is to learn a predictor f that is able to predict the labels Y of corresponding instances X for each $(X, Y) \sim \mathbb{P}(X, Y)$, given the finite training datasets drawn from the underlying distribution $\mathbb{P}(X, Y)$. However, unlike the standard supervised learning tasks, in domain generalization, we cannot sample directly from the distribution $\mathbb{P}(X, Y)$. Instead, we can only observe (X, Y) under different domains $e \in \mathcal{E}_{\text{all}}$, denoted as $(X^e, Y^e) \sim \mathbb{P}^e(X^e, Y^e)$. We also assume that $e \in \mathcal{E}_{\text{all}}$ is distributed as $e \sim \mathbb{P}_e$. Given samples from a finite subset $\mathcal{E}_{\text{train}} \subsetneq \mathcal{E}_{\text{all}}$ of all the domains, the goal of the domain generalization problem is to learn a predictor f that generalizes across all possible domains. This can be summarized as follows:

Problem 2.1 (Domain generalization)**.** *Let $\mathcal{E}_{train} \subsetneq \mathcal{E}_{all}$ be a finite subset of training domains. We have access to the data for each training domain $e_{train} \in \mathcal{E}_{train}$, but have no access to the data for each test domain $e_{test} \in \mathcal{E}_{all} \backslash \mathcal{E}_{train}$. Given a function class $\mathcal{F}$ and a loss function ℓ, our goal is to learn a predictor $f \in \mathcal{F}$ using the data from the training domain such that f minimizes the worst-case risk over $\mathcal{E}_{all}$. Define the risk of the predictor f on the domain e as $R^e(f) := \mathbb{E}_{\mathbb{P}^e(X^e, Y^e)}\, \ell(f(X^e), Y^e)$. We want to solve the following min-max optimization problem:*

$$\underset{f \in \mathcal{F}}{\text{minimize}} \max_{e \in \mathcal{E}_{all}} R^e(f). \tag{DG}$$

We establish the domain transformation model to characterize the relation between domain-aware instance X^e and the domain-invariant latent instance X in the Assumption 1, which first appears in (Robey et al., 2021).

Assumption 1 (Domain transformation model)**.** *Let δ_e denote a Dirac distribution for $e \in \mathcal{E}_{all}$. We assume that there exists a measurable function $G : \mathcal{X} \times \mathcal{E}_{all} \to \mathcal{X}$, which we refer to as a* domain transformation model*, that parameterizes the inter-domain covariate shift via*

$$\mathbb{P}^e(X) =^d G \# (\mathbb{P}(X) \times \delta_e) \quad \forall e \in \mathcal{E}_{all}, \tag{1}$$

where $\#$ denotes the push-forward measure and $=^d$ denotes equality in distribution.

Let Φ denote the feature representation mapping, w denote the classifier and $w \circ \Phi$ denote the full predictor. The regret of the network on the domain e is denoted as $R^e(w \circ \Phi)$.

Next, we define standard properties related to the datasets used in the domain generalization literature (Ahuja et al., 2021). For each $e \in \mathcal{E}_{\text{all}}$, the distribution $(X^e, Y^e) \sim \mathbb{P}^e$ satisfies the following properties: (1) $\exists$ a map Φ^*, which we call an *invariant feature map*, such that $\mathbb{E}\left[Y^e \middle| \Phi^*\left(X^e\right)\right]$ is the same for all $e \in \mathcal{E}_{all}$ and $Y^e \not\perp \Phi^*(X^e)$, where $\perp$ means mutual independence. (2) $\exists$ a map Ψ^*, which we call *spurious feature map*, such that $\mathbb{E}\left[Y^e \middle| \Psi^*\left(X^e\right)\right]$ is not the same for all $e \in \mathcal{E}_{all}$ and $Y^e \not\perp \Psi^*(X^e)$ for some domains. Ψ^* often hinders learning predictors that only rely on Φ^*. For example, in the CMNIST dataset, the $\Phi^\star$ extracts the underlying digit and $\Psi^\star$ extracts background color.

The baseline algorithm for domain generalization (DG) is the Empirical Risk Minimization, i.e. directly minimizing the empirical risk on the training domains:

$$\min_{w,\Phi} \frac{1}{|\mathcal{E}_{\text{train}}|} \sum_{e \in \mathcal{E}_{\text{train}}} R^e(w \circ \Phi), \tag{2}$$

where $|\mathcal{E}_{\text{train}}|$ denotes the number of training domains.

We say that a data representation Φ elicits an invariant predictor across the set of training domains $\mathcal{E}_{\text{train}}$ if there is a predictor w that simultaneously achieves the minimum risk, i.e. $w \in \arg\min_{w'} R^e(w' \circ \Phi), \forall e \in \mathcal{E}_{\text{train}}$. Using this notation, the main objective of Invariant Risk Minimization (IRM) is stated as:

$$\min_{w,\Phi} \frac{1}{|\mathcal{E}_{\text{train}}|} \sum_{e \in \mathcal{E}_{\text{train}}} R^e(w \circ \Phi), \qquad \text{s.t. } \; w \in \arg\min_{w'} R^e(w' \circ \Phi), \; \forall e \in \mathcal{E}_{\text{train}}. \tag{3}$$

Lastly, we rely on the notion of 'informativeness' about the datasets (Ahuja et al., 2021). There are two such categories of informativeness. In the first case, the invariant features $\Phi^*(X^e)$ are *partially informative* about the label, i.e. $Y \not\perp X^e | \Phi^*(X^e)$, and color contains information about label not contained in the uncolored digit. On the other hand in the second case, invariant features are *fully informative* about the label, i.e., $Y \perp X^e | \Phi^*(X^e)$, i.e., they contain all the information about the label that is contained in input X^e. Many real-world image datasets have fully informative invariant features, the labels are a deterministic function of the domain-invariant features and domain-aware spurious features do not affect the label.

3 METHOD

3.1 RATE DISTORTION & INFORMATION BOTTLENECK PRINCIPLE

Given a domain-free latent instance $X \in \mathbb{R}^{d_X}$, its observation in a domain e is denoted as $X^e := G(X, e)$. We want to learn the feature $Z^e = \Phi(X^e) \in \mathbb{R}^{d_Z}$ which is informative about the domain-free variable X, but invariant (i.e. uninformative) to the specific domain e. We use rate–distortion theory (Davisson, 1972; Blau & Michaeli, 2019) to formulate our domain generalization problem.

Rate–distortion theory is a major branch of information theory which provides the theoretical foundations for lossy data compression. An encoder Φ encodes domain-aware instances X^e. We want the representation $Z^e = \Phi(X^e)$ to be domain-invariant, so we feed Z^e into a decoder which outputs domain-invariant X. We minimize the distortion between the original domain-aware instance X^e and the reconstructed domain-free instance X. The distortion function measures how well X is predicted from a compressed representation Z^e compared to its direct prediction from X^e. This trade-off is captured by the following loss function:

$$L_{IB}(\theta, e) = \mathbb{E}_{X \sim \mathbb{P}_X, e \sim \mathbb{P}_e} I(Z^e; X^e) - \beta I(Z^e; X) \quad \text{IB objective} \tag{4}$$

where I denotes the mutual information, θ is the parameter of the representation function Φ, and β is a constant.

In the following, we consider two cases: discrete and continuous variables, owing to their different definition of entropy.

Discrete case: Since the representation function is deterministic with respect to θ, we can rewrite Equation (4) through a classical identity of mutual information: $I(X;Y) = H(X) - H(X|Y)$, where H denotes the Shannon entropy for discrete variables, as follows:

$$\begin{aligned} L_{IB}(\theta, e) &= \mathbb{E}_{X,e} I(Z^e; X^e) - \beta I(Z^e; X) \\ &= \mathbb{E}_{X,e} H(Z^e) - H(Z^e|X^e) - \beta(H(Z^e) - H(Z^e|X)) \\ &= \mathbb{E}_{X,e} H(Z^e|X) + \frac{1-\beta}{\beta} H(Z^e), \end{aligned} \tag{5}$$

where in the last equality we omit the overall scaling factor of the loss function.

If $0 \leq \beta \leq 1$, since $H(\cdot)$ is bounded below by 0, setting Φ to be constant will clearly minimize the penalty, which is uninformative about the representations we want to learn. Hence, we set $\beta > 1$, and replace $\frac{1-\beta}{\beta}$ with $-\lambda$, where $0 \leq \lambda < 1$. The IB objective can be rewritten as

$$L_{IB}(\theta, e) = \mathbb{E}_{X,e} H(Z^e|X) - \lambda H(Z^e). \tag{6}$$

Continuous case: In terms of continuous variables, the differential entropy $h(\cdot)$ is not bounded below, which hinders our analysis. To overcome this, we can define the lower bounded differential entropy $\widehat{h}(X) := h(X + \varepsilon)$, where ε is the independent bounded zero-entropy noise $\varepsilon \sim \mathrm{Uniform}(0, 1)$. Thus, $\widehat{h}(X) \geq h(\varepsilon) = 0$. We can replace the Shannon entropy $H(\cdot)$ with lower bounded differential entropy $\widehat{h}(\cdot)$ in Equation (6):

$$L_{IB}(\theta, e) = \mathbb{E}_{X,e}\widehat{h}(Z^e|X) - \lambda\widehat{h}(Z^e) \tag{7}$$

For the discrete case, we define $H^e(f) := \mathbb{E}_{X^e\sim\mathbb{P}^e} H(f(X^e))$ and $\widetilde{H}^e(f) := \mathbb{E}_{X^e\sim\mathbb{P}^e} H(f(X^e)|X)$. For the continuous case, we define $\widehat{h}^e(f) := \mathbb{E}_{X^e\sim\mathbb{P}^e}\widehat{h}(f(X^e))$ and $\widetilde{h}^e(f) := \mathbb{E}_{X^e\sim\mathbb{P}^e}\widehat{h}(f(X^e)|X)$. Using this terminology, we are able to incorporate the IB penalty into the Empirical Risk Minimization (ERM) algorithm, denoted as Twins-ERM method,

$$\begin{aligned}
&\min_{w,\Phi} \sum_{e\in\mathcal{E}_{\text{train}}} \widetilde{H}^e(\Phi) - \lambda H^e(\Phi) \quad \text{s.t.} \ \frac{1}{|\mathcal{E}_{\text{train}}|}\sum_{e\in\mathcal{E}_{\text{train}}} R^e\big(w\circ\Phi\big) \leq r, \ \text{(discrete case)} \\
&\min_{w,\Phi} \sum_{e\in\mathcal{E}_{\text{train}}} \widetilde{h}^e(\Phi) - \lambda \widehat{h}^e(\Phi) \quad \text{s.t.} \ \frac{1}{|\mathcal{E}_{\text{train}}|}\sum_{e\in\mathcal{E}_{\text{train}}} R^e\big(w\circ\Phi\big) \leq r, \ \text{(continuous case)}
\end{aligned} \tag{8}$$

where r is the threshold on the empirical risk on the training domains.

In addition to ERM, another popular minimization framework is the invariant risk minimization (Arjovsky et al., 2019). The proposed penalty can be readily incorporated into the IRM framework, we call the resulting algorithm Twins-IRM:

$$\begin{aligned}
&\min_{w,\Phi} \sum_{e\in\mathcal{E}_{\text{train}}} \widetilde{H}^e(\Phi) - \lambda H^e(\Phi) \ \text{(discrete case)}, \quad \text{or,} \sum_{e\in\mathcal{E}_{\text{train}}} \widetilde{h}^e(\Phi) - \lambda h^e(\Phi) \ \text{(continuous case)} \\
&\text{s.t.} \ \frac{1}{|\mathcal{E}_{\text{train}}|}\sum_{e\in\mathcal{E}_{\text{train}}} R^e\big(w\circ\Phi\big) \leq r, \ w \in \arg\min_{\tilde{w}} R^e\big(\tilde{w}\circ\Phi\big), \forall e \in \mathcal{E}_{\text{train}}.
\end{aligned} \tag{9}$$

3.2 Theoretical Guarantee

In this subsection, we establish the theoretical guarantee of our algorithm under the linear classification case. We consider the following standard 0-1 classification model in literature (Ahuja et al., 2020; 2021):

Assumption 2. *Linear classification structural equation model. In each $e \in \mathcal{E}_{all}$,*

$$\begin{aligned}
&Y^e \leftarrow \mathrm{I}(w^\star_{\text{inv}} \cdot X_{\text{inv}}) \oplus N^e, \quad N^e \sim \text{Bernoulli}(q), q \leq \frac{1}{2}, \quad N^e \perp (X_{\text{inv}}, X^e_{\text{spu}}) \\
&X^e \leftarrow S(X^e_{\text{inv}}, X^e_{\text{spu}}), \quad X^e_{\text{inv}} \leftarrow G(X_{\text{inv}}, e)
\end{aligned} \tag{10}$$

where $\mathrm{I}(x)$ is 1 if x is positive else 0, $w^\star_{\text{inv}} \in \mathbb{R}^m$ with $\|w^\star_{\text{inv}}\| = 1$ is the labelling hyperplane, $X^e_{\text{inv}}, X^e_{\text{inv}} \in \mathbb{R}^m, X^e_{\text{spu}} \in \mathbb{R}^o$, $S \in \mathbb{R}^{(m+o)\times(m+o)}$ and G is a continuous domain transformation.

Before presenting our main theorem, we first add two assumptions on the support of invariant features. Define the support of the invariant features X^e_{inv} in environment e as $\mathcal{X}^e_{\text{inv}}$.

Assumption 3 (Invariant feature support overlap)**.** *The union of support of the invariant features of the training domains covers support of the invariant features of all the domains. i.e. $\bigcup_{e\in\mathcal{E}_{all}} \mathcal{X}^e_{\text{inv}} \subseteq \bigcup_{e\in\mathcal{E}_{train}} \mathcal{X}^e_{\text{inv}}$.*

Assumption 4 (Strictly separable invariant features)**.** *The training support of invariant features $\bigcup_{e\in\mathcal{E}_{train}} \mathcal{X}^e_{\text{inv}}$ is strictly separated by the labelling hyperplane w^*_{inv} into two halves.*

Theorem 3.1. *Suppose each $e \in \mathcal{E}_{all}$ follows Assumption 2, and Assumptions 3 and 4 hold for the invariant features. Also, for each $e \in \mathcal{E}_{train}$, assume that $X^e_{\text{spu}} = AX^e_{\text{inv}} + W^e$, where $A \in \mathbb{R}^{o\times m}$, $W^e \in \mathbb{R}^o$ is continuous, bounded, and zero mean noise. Each solution to Twins-ERM and Twins-IRM (Equation (8) and Equation (9), with ℓ as 0-1 loss, and $r = q$) solves the domain generalization problem (Equation (DG)).*

Sketch of Proof. The full proof is provided at Appendix A. We only present the main idea here. Denote $\Phi^\dagger$ as the solution to Equation (9),

$$\Phi^\dagger X^e = \Phi^\dagger S(X^e_{\text{inv}}, X^e_{\text{spu}}) = \Phi^\dagger_{\text{inv}} X^e_{\text{inv}} + \Phi^\dagger_{\text{spu}} X^e_{\text{spu}} = (\Phi^\dagger_{\text{inv}} + \Phi^\dagger_{\text{spu}} \cdot A) X^e_{\text{inv}} + \Phi^\dagger_{\text{spu}} W^e \quad (11)$$

We will show that $\Phi^+ = \left(\left[\Phi_{\text{inv}} + \Phi_{\text{spu}} \cdot A\right], 0\right) S^{-1}$ can continue to achieve an error of $q(= r)$ across training domains, and have a lower information bottleneck penalty. Therefore, the optimal solution to Twins-ERM (Equation (8)) does not depend on the spurious noise W^e, and hence solves the domain generalization problem (Equation (DG)). □

It is known that ERM and IRM fails under the assumption of Theorem 3.1 (Theorem 3 in (Ahuja et al., 2021)). This theorem shows that our algorithm can provably solve the linear classification structural equation model.

3.3 Gaussian Bottleneck

In order to facilitate the implementation of our penalty, we make the simplifying assumption that the datasets are distributed as Gaussian (Chechik et al., 2005). To be more specific, assuming X, X^e are jointly multivariate zero-mean Gaussian vectors with covariances Σ_X, Σ_{X^e}, and $Z^e \in \mathbb{R}^{d_Z}$ is a encoded version of X^e that must maintain a given value of mutual information with X. We define the covariance of Z^e and $Z^e|X$ to be Σ_{Z^e} and $\Sigma_{Z^e|X}$.

Next, we simplify the Equation (6). The entropy of a Gaussian distribution is simply given by the logarithm of the determinant of its covariance function (up to a constant that we ignore). The loss function becomes:

$$L_{IB}(\theta, e) = \mathbb{E}_X \log\det(\Sigma_{Z^e|X}) - \lambda \log\det(\Sigma_{Z^e}). \quad (12)$$

Practical considerations: We reformulate our algorithm so that it resembles the contrastive learning method, i.e. Barlow Twins (Zbontar et al., 2021). The second term of the loss in Equation (12) maximizes $\det(\Sigma_{Z^e})$. Since computing the determinant of a matrix is computationally intensive, we adopt a proxy to minimize the Frobenius norm of the correlation matrix of Z^e. Since the correlation matrix is invariant to scaling, we can set the diagonal element of the correlation matrix of Z^e to be 1. Then, the second term of Equation (12) amounts to minimizing the off-diagonal term, i.e. the second term in Equation (14). This term essentially decorrelates the different dimensions of the representation and prevents these dimensions from encoding similar information.

Besides, it can also easily be shown that the first term of Equation (12) minimizes the information the representation contains about the domain information has the same solution with the first term of Equation (14). This term maximizes the alignment between representations of pairs of domain-aware instances X^e and domain-free instances X.

In practice, we have *no access to the domain-free latent instance*. For a given instance X^e, we use an instance with the same label, but from a different domain, denoted as $X^{e'}$ as surrogate for the domain-free latent X. We minimize the distance between pairs of instances from different domains. Sample $\{z^e_b\}_{1\le b\le B} \sim Z^e$ and $\{z^{e'}_b\}_{1\le b\le B} \sim Z^{e'}$, we concatenate them into matrix $\mathbf{Z}^e \in \mathbb{R}^{B\times d_Z}$ and $\mathbf{Z}^{e'} \in \mathbb{R}^{B\times d_Z}$, where d_Z is the dimension of $z^e_b, z^{e'}_b$. The cross-correlation matrix C^Z_{ij} is defined as:

$$C^Z_{ij} = \frac{\langle \mathbf{Z}^e_{:,i}, \mathbf{Z}^{e'}_{:,j} \rangle}{\|\mathbf{Z}^e_{:,i}\|_2 \|\mathbf{Z}^{e'}_{:,j}\|_2}, 1 \le i, j \le d_Z, \quad (13)$$

and the final penalty is defined as:

$$c(Z) = \sum_i (1 - C^Z_{ii})^2 + \lambda \sum_{i\neq j} (C^Z_{ij})^2. \quad (14)$$

However, we do not have access to domain transformation model either. We construct the contrastive instances by permuting the instances that have the same label in each iteration. In particular, we sample B instances $\{x_b\}_{1\le b\le B}$ from training domains as row vectors, where B is the batch size. We concatenate the representation $\{z_b\} = \{\Phi(x_b)\}$ into matrix $\mathbf{Z}^1 \in \mathbb{R}^{B\times d_Z}$. The contrastive

batch $\mathbf{Z}^2$ is constructed by permuting the rows of $\mathbf{Z}^1$, i.e. $Z^1_{b,:} = Z^2_{\pi(b),:}$, where π is a permutation of $\{1, 2, \cdots, B\}$ such that the corresponding labels of x_b and $x_{\pi(b)}$ are identical. C^Z_{ij} defined in Equation (13) can be rewritten as:

$$C^Z_{ij} = \frac{\langle \mathbf{Z}^1_{:,i}, \mathbf{Z}^2_{:,j} \rangle}{\|\mathbf{Z}^1_{:,i}\|_2 \|\mathbf{Z}^2_{:,j}\|_2}, 1 \leq i, j \leq d_Z \tag{15}$$

Such penalty can be readily incorporated into the ERM and IRM losses, i.e.

$$L_{\text{Twins}-\text{ERM}} = L_{\text{ERM}} + \mu \cdot c(Z), \text{ and } L_{\text{Twins}-\text{IRM}} = L_{\text{IRM}} + \mu \cdot c(Z), \tag{16}$$

where L_{ERM} and L_{IRM} denote the loss in the ERM and IRM respectively, and μ is the penalty hyperparameter.

4 Experiments

In this section, we conduct experimentation on two benchmarks: Linear Unit Tests (in Section 4.1) and DomainBed (in Section 4.2). Linear Unit Tests (Aubin et al., 2021) consists of several toy datasets to evaluate algorithms for domain generalization and invariance learning, while DomainBed (Gulrajani & Lopez-Paz, 2020) is a unified testbed useful for evaluating domain generalization algorithms. We use the following *four baselines* across our experiments: ERM, IB-ERM, IRM, IB-IRM (Ahuja et al., 2021).

4.1 Linear Unit Tests

The dataset describes six linear low-dimensional problems, named Example 1/1s, Example 2/2s and Example 3/3s, where the 's' dictates a different rotation matrix. Each example, called unit test, is designed to test different types of out-of-distribution generalization. We describe below the precise distributions and the invariances captured by each example.

Example 1/1s The dataset in environment $e \in \mathcal{E}_{\text{all}}$ is sampled from the following

$$\begin{aligned}
&X^e_{\text{inv}} \sim \mathcal{N}_{d_{\text{inv}}}(0, (\sigma^e)^2), \qquad \tilde{Y}^e \sim \mathcal{N}_{d_{\text{inv}}}(W_{yx} X^e_{\text{inv}}, (\sigma^e)^2), \\
&X^e_{\text{spu}} \sim \mathcal{N}_{d_{\text{spu}}}(W_{xy} \tilde{Y}^e, 1), \quad X^e \leftarrow S \cdot (X^e_{\text{inv}}, X^e_{\text{spu}}), \\
&Y^e \leftarrow \frac{2}{(d_{\text{inv}} + d_{\text{spu}})} \mathbf{1}^{\text{T}}_{d_{\text{inv}}} \tilde{Y}^e,
\end{aligned} \tag{17}$$

where $W_{yz} \in \mathbb{R}^{d_{\text{inv}} \times d_{\text{inv}}}$, $W_{xy} \in \mathbb{R}^{d_{\text{spu}} \times d_{\text{inv}}}$ are matrices drawn i.i.d. from the standard normal distribution, $\mathbf{1}_{d_{\text{inv}}} \in \mathbb{R}^{d_{\text{inv}}}$ is a vector of ones, $\mathcal{N}_k$ is a k dimensional vector from the normal distribution, and $S \in \mathbb{R}^{(d_{\text{inv}}+d_{\text{spu}}) \times (d_{\text{inv}}+d_{\text{spu}})}$ is a rotation matrix fixed for all environments. The parameter σ is set differently for every environment (i.e., domain). In particular, we set $(\sigma^{e=e_0})^2 = 0.1$, $(\sigma^{e=e_1})^2 = 1.5$, and $(\sigma^{e=e_2})^2 = 2$ for the first three environments. In case, there are more than three environments, the $(\sigma^{e=e_j})$ for $j > 3$ is uniformly from $\text{Unif}(10^{-2}, 10)$. The rotation matrix S is set to the identity matrix in Example 1 and a random unitary matrix in Example 1s.

Example 2/2s Following the notation of the original paper Aubin et al. (2021), let

$$\begin{aligned}
&\mu_{\text{cow}} = \mathbf{1}_{\text{d}_{\text{inv}}}, \qquad \mu_{\text{camel}} = -\mu_{\text{cow}}, \qquad \nu_{\text{animal}} = 10^{-2}, \\
&\mu_{\text{grass}} = \mathbf{1}_{\text{d}_{\text{spu}}}, \qquad \mu_{\text{sand}} = -\mu_{\text{grass}}, \qquad \nu_{\text{background}} = 1.
\end{aligned} \tag{18}$$

The dataset in environment $e \in \mathcal{E}_{\text{all}}$ is sampled from the following distribution:

$$\begin{aligned}
&j^e \sim \text{Categorical}\big(p^e s^e, (1-p^e)s^e, p^e(1-s^e), (1-p^e)(1-s^e)\big), \\
&X^e_{\text{inv}} \sim \begin{cases} (\mathcal{N}_{d_{\text{inv}}}(0, 0.1) + \mu_{\text{cow}}) \cdot \nu_{\text{animal}} & \text{if } j^e \in \{1, 2\}, \\ (\mathcal{N}_{d_{\text{inv}}}(0, 0.1) + \mu_{\text{camel}}) \cdot \nu_{\text{animal}} & \text{if } j^e \in \{3, 4\}, \end{cases} \\
&X^e_{\text{spu}} \sim \begin{cases} (\mathcal{N}_{d_{\text{spu}}}(0, 0.1) + \mu_{\text{grass}}) \cdot \nu_{\text{background}} & \text{if } j^e \in \{1, 4\}, \\ (\mathcal{N}_{d_{\text{spu}}}(0, 0.1) + \mu_{\text{sand}}) \cdot \nu_{\text{background}} & \text{if } j^e \in \{2, 3\}, \end{cases} \\
&X^e \leftarrow S \cdot (X^e_{\text{inv}}, X^e_{\text{spu}}), \quad Y^e \leftarrow \text{I}(\mathbf{1}^{\text{T}}_{d_{\text{inv}}} X^e_{\text{inv}}),
\end{aligned} \tag{19}$$

where the environment foreground/background probabilities are $p^{e=e_0} = 0.95$, $p^{e=e_1} = 0.97$, $p^{e=e_2} = 0.99$ and the cow/camel probabilities are $s^{e=e_0} = 0.3$, $s^{e=e_1} = 0.5$, $s^{e=e_2} = 0.7$. For $n_{\text{env}} > 3$ and $j \in [3 : n_{\text{env}} - 1]$, the extra environment variables are respectively drawn according to $p^{e=e_j} \sim \text{Unif}(0.9, 1)$ and $s^{e=e_j} \sim \text{Unif}(0.3, 0.7)$. The rotation matrix S is set to the identity matrix in Example 2 and a random unitary matrix in Example 2s.

Example 3/3s The example is meant to present a linear version of the spiral classification problem of Parascandolo et al. (2020). Let $\mu_{\text{inv}} = 0.1 \cdot \mathbf{1}_{d_{\text{inv}}}$, and $\mu^e_{\text{spu}} \sim \mathcal{N}_{d_{\text{spu}}}(0, 1)$ for all the environments. The dataset in environment $e \in \mathcal{E}_{\text{all}}$ is sampled from the following distribution:

$$Y^e \sim \text{Bernoulli}\Big(\frac{1}{2}\Big), \qquad X^e \leftarrow S \cdot (X^e_{\text{inv}}, X^e_{\text{spu}})$$
$$X^e_{\text{inv}} \sim \begin{cases} \mathcal{N}_{d_{\text{inv}}}(+\mu_{\text{inv}},\ 0.1) \text{ if } Y^e = 0, \\ \mathcal{N}_{d_{\text{inv}}}(-\mu_{\text{inv}},\ 0.1) \text{ if } Y^e = 1, \end{cases} \qquad X^e_{\text{spu}} \sim \begin{cases} \mathcal{N}_{d_{\text{spu}}}(+\mu^e_{\text{spu}},\ 0.1) \text{ if } Y^e = 0, \\ \mathcal{N}_{d_{\text{spu}}}(-\mu^e_{\text{spu}},\ 0.1) \text{ if } Y^e = 1, \end{cases} \tag{20}$$

The rotation matrix S is set to the identity matrix in Example 3 and a random unitary matrix in Example 3s. In the above dataset, the invariant features are anti-causally related to the label Y^e.

Remark on Linear unit test: In the Example 1/1s and Example 3/3s, the invariant features are causal and partially informative about the label. The spurious features carry extra information about the label not contained in the invariant features. In the Example 2/2s, the invariant features are causal and carry full information about the label.

Benchmark details: We follow the same pipeline as those used in (Aubin et al., 2021; Ahuja et al., 2021) for the model selection, hyperparameter selection, training, and evaluation. We set $(d_{\text{inv}}, d_{\text{spu}}) = (5, 5)$. For all three examples, the models used are linear. The training loss is the square error for the regression setting (Example 1/1s), and binary cross-entropy for the classification setting (Example 2/2s, 3/3s). For the evaluation of performance on Example 1/1s, we report mean square errors and standard deviations. For the evaluation of performance on Example 2/2s, Example 3/3s, we report classification errors and standard deviations.

Model training: For the Twins-ERM approach, there is an additional hyperparameter μ associated with the $c(Z)$ term in the final objective in Equation (16). We sample the μ from $\log \mu \sim \text{Uniform}(-3, -1)$. For each algorithm, we run a random hyperparameter search for 20 trials, and average the results over 50 data seeds. We train each algorithm and hyperparameter trial on the train splits of all environments, for 10^4 full-batch Adam updates (Kingma & Ba, 2014). We choose the hyperparameters trial that minimizes the error on the validation splits of all environments, i.e. the train-domain validation set evaluation procedure in (Gulrajani & Lopez-Paz, 2020). We also implement an Oracle that contains randomized X_{spu} in each iteration, such that it learns to ignore the spurious features.

	#Envs	ERM	IB-ERM	IRM	IB-IRM	Twins-ERM	Twins-IRM	Oracle
Example1	3	13.36 ± 1.49	12.96 ± 1.30	$\mathbf{11.15 \pm 0.71}$	11.68 ± 0.90	14.62 ± 1.20	14.42 ± 0.86	10.42 ± 0.16
Example1s	3	13.33 ± 1.49	12.92 ± 1.30	$\mathbf{11.07 \pm 0.68}$	11.74 ± 1.03	14.64 ± 1.22	13.25 ± 1.49	10.45 ± 0.19
Example2	3	0.42 ± 0.01	0.00 ± 0.00	0.45 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.00	0.00 ± 0.00
Example2s	3	0.45 ± 0.01	0.00 ± 0.01	0.45 ± 0.01	0.06 ± 0.12	$\mathbf{0.00 \pm 0.00}$	0.43 ± 0.03	0.00 ± 0.00
Example3	3	0.48 ± 0.07	0.49 ± 0.06	0.48 ± 0.07	0.48 ± 0.07	$\mathbf{0.42 \pm 0.15}$	0.33 ± 0.14	0.00 ± 0.00
Example3s	3	0.49 ± 0.06	0.49 ± 0.06	0.49 ± 0.07	0.49 ± 0.07	0.50 ± 0.05	$\mathbf{0.42 \pm 0.11}$	0.00 ± 0.00
Example2	6	0.37 ± 0.06	0.02 ± 0.05	0.46 ± 0.01	0.43 ± 0.11	$\mathbf{0.00 \pm 0.00}$	0.07 ± 0.11	0.00 ± 0.00
Example2s	6	0.46 ± 0.01	0.02 ± 0.06	0.46 ± 0.01	0.45 ± 0.10	$\mathbf{0.00 \pm 0.00}$	0.47 ± 0.00	0.00 ± 0.00
Example3	6	0.33 ± 0.18	0.26 ± 0.20	$\mathbf{0.14 \pm 0.18}$	0.19 ± 0.19	0.24 ± 0.16	0.25 ± 0.20	0.01 ± 0.00
Example3s	6	0.36 ± 0.19	0.27 ± 0.20	$\mathbf{0.14 \pm 0.18}$	0.19 ± 0.19	0.31 ± 0.19	0.44 ± 0.06	0.01 ± 0.00

Table 1: Comparisons on linear unit tests in terms of mean square error (regression, ↓) in Example 1/1s and classification error (classification, ↓) in Examples 2/2s and 3/3s. The highlighted result per example demonstrates the best performance. When #Envs=6, we do not report results on Example 1/1s, since even the oracle cannot obtain stable results across different data seeds.

The experimental results are reported in Table 1. In the Example 2/2s, since the invariant feature contains full information about the label, we do observe that IB penalty in (Ahuja et al., 2021) and Twins penalty in our paper performs the best. In the Example 1/1s and 3/3s, the spurious feature

contains partial information about the label, we generally find invariant risk can reduce the error in this case. We empirically verify the benefit of using the proposed penalty over the previously proposed baselines.

4.2 MNIST-TYPE DATASET

In the second benchmark we use DomainBed to experiment on MNIST-type datasets inspired by the construction of CS-CMNIST (Ahuja et al., 2021) to evaluate covariate shift. In addition to the origin MNIST (LeCun & Cortes, 2010), we extend the benchmark to include FashionMNIST (Xiao et al., 2017) and KMNIST (Clanuwat et al., 2018), where the images of the latter two are used as drop-in replacements for MNIST images. The idea in CS-CMNIST is to associate each class with a color, and each image is assigned the color associated to its class with probability p^e or a random color with probability $1 - p^e$. We construct three environments for this experiment: two training environments containing 20,000 data points each, one test containing 20,000 points. In the two training environments, the p^e is set to 1.0 and 0.9 respectively. In the testing environment, the p^e is set to 0, i.e., all the images are colored completely at random. A grid search in the range of $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$ is used to determine the optimal penalty parameter μ. We fix the trade-off parameter $\lambda = 5 \times 10^{-3}$. We run the experiments using 5 different seeds and report the mean and the standard deviation of the classification.

The results are reported in Table 2. Notice that both the IRM and the ERM versions in each case perform similarly. The results reveal that ERM and IRM have the weakest performance. IB-ERM and IB-IRM increase the accuracy of ERM and IRM respectively, with Twins-ERM and Twins-IRM outperforming all the compared methods.

	ERM	IB-ERM	IRM	IB-IRM	Twins-ERM	Twins-IRM
MNIST	60.27 ± 1.21	71.80 ± 0.69	61.49 ± 1.45	71.79 ± 0.70	$\mathbf{83.03 \pm 1.34}$	82.83 ± 2.73
FashionMNIST	50.92 ± 1.20	51.74 ± 1.12	48.41 ± 0.90	50.92 ± 1.20	55.60 ± 3.33	$\mathbf{56.04 \pm 1.79}$
KMNIST	22.80 ± 1.06	29.21 ± 0.85	22.89 ± 0.94	27.83 ± 0.37	51.24 ± 3.94	$\mathbf{51.52 \pm 3.83}$

Table 2: Classification accuracy (↑) on MNIST-type datasets. Notice that the proposed Twins-ERM and Twins-IRM exhibit the best performance outperforming previous methods by a significant margin.

5 RELATED WORK

5.1 RELATION TO INVARIANT RISK MINIMIZATION

Relation to IB-ERM/IB-IRM (Ahuja et al., 2021). Ahuja et al. (2021) introduce the information bottleneck method. However, similar to Tishby et al. (2000), Ahuja et al. (2021) utilize the information bottleneck to learn a representation that compresses the input as much as possible while preserving all the relevant information about the target label. Ours instead is label-free and tries to preserve the relevant information about the domain-free latent image during compression, and uses another image with the same label but different domain as surrogate for the unknown domain-free latent image. Essentially, setting $\beta = 0$ will reduce our penalty into the one in Ahuja et al. (2021), but $\beta > 1$ can help eliminate the trivial solution that the representation mapping is constant, and is shown to achieve better performance on various datasets.

Previous work has demonstrated that the entropy penalty alone (i.e. $\beta = 0$) might fail in specific case, such as in Section 4 in Ahuja et al. (2021). Nevertheless, our proposed framework does not suffer from this counter-example. We introduce such an example next as a simple classification problem. In each $e \in \mathcal{E}_{\text{train}}$, $Y^e \leftarrow X^e_{\text{inv}} \oplus N^e$ and $X^e_{\text{spu}} \leftarrow Y^e \oplus V^e$, where all the random variables involved are binary valued, noise N^e, V^e are Bernoulli with parameters q (identical across $\mathcal{E}_{\text{train}}$), c^e (varies across $\mathcal{E}_{\text{train}}$) respectively. If $c^e < q$, then in $\mathcal{E}_{\text{train}}$ predictions based on X^e_{spu} are better than predictions based on X^e_{inv}. If Φ selects X^e_{inv}, the IB penalty equals $-\lambda H(X^e_{\text{inv}})$; while if Φ selects X^e_{spu}, the IB penalty equals $H(N^e \oplus V^e) - \lambda H(X^e_{\text{spu}})$. Since $X^e_{\text{spu}} \leftarrow X^e_{\text{inv}} \oplus N^e \oplus V^e$, we have $-\lambda H(X^e_{\text{inv}}) < H(N^e \oplus V^e) - \lambda H(X^e_{\text{spu}})$ by Lemma A.1 in the Appendix. Our IB penalty is then able to select the invariance term X^e_{inv}. On the other hand, if X^e_{inv} obeys uniform Bernoulli distribution, its entropy will be no lower than X^e_{spu}, and hence the entropy term alone is not enough.

5.2 Relation to Penalty-based Domain Generalization

CORAL (Sun & Saenko, 2016) explores a covariance term that bears resemblance with our regularization scheme. Given two representations $Z^e, Z^{e'}$ from different domains e and e', CORAL (Sun & Saenko, 2016) aligns the correlation matrix from different domains, while our method (Equation (14)) decorrelates each dimension of the representation by minimizing the off-diagonal term of cross-correlation matrix. The motivation and implication of the proposed regularization scheme are different and our method can provide a guaranteed convergence to the invariant feature.

Our method can also be regarded as a contrastive-based domain generalization problem. Contrastive learning (Chopra et al., 2005; Caron et al., 2020; Grill et al., 2020; Chen et al., 2020; Zbontar et al., 2021) has been a successful paradigm in self-supervised leaning. Contrastive learning aims at bringing positive pair samples closer together, while moving negative samples further away in a learned embedding space. Essentially, the aim of domain generalization is to extract domain-invariant features, similarly aiming to minimize the distance of features within the same class in the embedding space, while maximizing the distance of features from different classes. However, explicit construction of negative pairs often hinder the training since informative negative samples are challenging to construct. In order to design an algorithm free of negative samples, our goal is to make network parameters converge to domain-invariant features and prevent the representation collapse (Grill et al., 2020). SelfReg (Kim et al., 2021) uses several gradient stabilization techniques from literature on contrastive learning. Our method instead introduces a much simpler penalty to achieve such goal with theoretical guarantee.

6 Conclusion

In this work, we introduce an information-theoretical approach for domain generalization. We cast the task of domain generalization as a rate distortion problem and then use information bottleneck penalty to obtain guarantees on the existence of features we want to learn. We link our method, called Twins, with self-supervised learning, which can provide a theoretical perspective in the success behind self-supervised learning. We conduct an empirical study on Twins-ERM and Twins-IRM under various datasets and confirm the consistent improvement of the proposed method over existing baselines. In the future, we intend to further analyze domain generalization in the rate distortion framework.

References

Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*, 2020.

Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *arXiv preprint arXiv:2106.06607*, 2021.

Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4845–4854, 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Benjamin Aubin, Agnieszka Słowik, Martin Arjovsky, Leon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31: 998–1008, 2018.

Normand J Beaudry and Renato Renner. An intuitive proof of the data processing inequality. *arXiv preprint arXiv:1107.0740*, 2011.

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

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24:2178–2186, 2011.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pp. 675–685. PMLR, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Gal Chechik, Amir Globerson, Naftali Tishby, Yair Weiss, and Peter Dayan. Information bottleneck for gaussian variables. *Journal of machine learning research*, 6(1), 2005.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

L Davisson. Rate distortion theory: A mathematical basis for data compression. *IEEE Transactions on Communications*, 20(6):1202–1202, 1972.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Daehee Kim, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. *arXiv preprint arXiv:2104.09841*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andreas Kirsch, Clare Lyle, and Yarin Gal. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning. *arXiv preprint arXiv:2003.12537*, 2020.

Jaihyun Koh, Jangho Lee, and Sungroh Yoon. Single-image deblurring with neural networks: A comparative survey. *Computer Vision and Image Understanding*, 203:103134, 2021.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b.

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)*, pp. 624–639, 2018c.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Alexander Robey, George J Pappas, and Hamed Hassani. Model-based domain generalization. *arXiv preprint arXiv:2102.11436*, 2021.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Alfred Wehrl. General properties of entropy. *Reviews of Modern Physics*, 50(2):221, 1978.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pp. 561–578. Springer, 2020.

A PROOF OF THEOREM 3.1

The entropy or the Shannon entropy (Wehrl, 1978) of a discrete random variable $X \sim \mathbb{P}_X$ with support $\mathcal{X}$ is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} \mathbb{P}_X(X = x) \log\big(\mathbb{P}_X(X = x)\big). \tag{21}$$

The differential entropy (Wehrl, 1978) of a continuous random variable $X \sim \mathbb{P}_X$ with support $\mathcal{X}$ is given as follows

$$h(X) = -\int_{x \in \mathcal{X}} \log\big(\mathbb{P}_X(x)\big) d\mathbb{P}_X(x), \tag{22}$$

where $d\mathbb{P}_X(x)$ is the Radon-Nikodym derivative of $\mathbb{P}_X$ w.r.t the Lesbegue measure.

For continuous variables, the differential entropy $h(\cdot)$ is not bounded below, we can define the lower bounded differential entropy (Kirsch et al., 2020) $\widehat{h}(X) = h(X + \varepsilon)$, where ε is an independent zero-entropy noise $\varepsilon \sim \text{Uniform}(0, 1)$. Since $X \perp \varepsilon$, $\widehat{h}(X) \geq h(\varepsilon) = 0$, we get that $\widehat{h}(\cdot)$ is bounded below.

Lemma A.1. *If X and Y are discrete random variables that are independent with the supports satisfying $2 \leq |\mathcal{X}| < \infty, 2 \leq |\mathcal{Y}| < \infty$, where $|\cdot|$ denotes the number of element in a set, then for $\lambda < 1$,*

$$\lambda H(X) + H(Y) > \lambda H(X + Y) \tag{23}$$

Proof. Define $Z = X + Y$.

$$\begin{aligned}
H(Z|X) &= -\sum_{x \in \mathcal{X}} \mathbb{P}_X(x) \sum_{z \in \mathcal{Z}} \mathbb{P}_{Z|X}(Z = z|X = x) \log\Big(\mathbb{P}_{Z|X}(Z = z|X = x)\Big) \\
&= -\sum_{x \in \mathcal{X}} \mathbb{P}_X(x) \sum_{z \in \mathcal{Z}} \mathbb{P}_{Y|X}(Y = z - x|X = x) \log\Big(\mathbb{P}_{Y|X}(Y = z - x|X = x)\Big) \\
&= -\sum_{x \in \mathcal{X}} \mathbb{P}_X(x) \sum_{z \in \mathcal{Z}} \mathbb{P}_{Y|X}(Y = z - x|X = x) \log\Big(\mathbb{P}_{Y|X}(Y = z - x|X = x)\Big) \text{ (use } X \perp Y) \\
&= -\sum_{x \in \mathcal{X}} \mathbb{P}_X(x) \sum_{z \in \mathcal{Z}} \mathbb{P}_Y(Y = z - x) \log\Big(\mathbb{P}_Y(Y = z - x)\Big) \\
&= H(Y)
\end{aligned} \tag{24}$$

Hence,

$$H(X+Y) - H(X) = H(X+Y) - H(X+Y|Y) = I(X+Y;Y) = H(Y) - H(Y|X+Y) \leq H(Y) \tag{25}$$

$$\lambda(H(X + Y) - H(X)) \leq \lambda H(Y) \leq H(Y) \tag{26}$$

when $\lambda < 1$.

The equality holds if and only if $H(Y) = 0$, which is impossible since $2 \leq |\mathcal{Y}| < \infty$. □

Lemma A.2. *If X and Y are continuous random variables that are independent and have a bounded support, then for $\lambda < 1$,*

$$\widehat{h}(X) + \lambda \widehat{h}(Y) > \lambda \widehat{h}(X + Y) \tag{27}$$

Proof. Setting $\varepsilon \sim \text{Uniform}(0, 1)$ independent of X, Y, we have $\widehat{h}(X) = h(X + \varepsilon), \widehat{h}(Y) = h(Y + \varepsilon), \widehat{h}(X + Y) = h(X + Y + \varepsilon)$. We have

$$\begin{aligned}
\lambda(h(X + Y + \varepsilon) - h(Y + \varepsilon)) &= \lambda I(X + Y + \varepsilon; X) \\
\text{and } h(X + \varepsilon) = h(X + \varepsilon) - h(\varepsilon) &= I(X + \varepsilon; X)
\end{aligned} \tag{28}$$

According to the data processing inquality (Beaudry & Renner, 2011),

$$I(X+Y+\varepsilon;X) \leq I(X+\varepsilon,Y;X) = I(X+\varepsilon;X) \tag{29}$$

where the last equality holds since $X+\varepsilon \perp Y$, and we get

$$\lambda(h(X+Y+\varepsilon)-h(Y+\varepsilon)) \leq I(X+Y+\varepsilon;X) \leq I(X+\varepsilon;X) = h(X+\varepsilon) \tag{30}$$

Rearranging it, we get

$$\widehat{h}(X)+\lambda\widehat{h}(Y) \geq \lambda\widehat{h}(X+Y) \tag{31}$$

Since $\lambda < 1$, the equality holds only if $I(X+Y+\varepsilon;X)=0$. In other words, we have $X+Y+\varepsilon \perp X$. In the next, we show that it is not possible.

The support of X can be divided into the union of intervals. We assume $\Delta > 0$ such that $[x_{\max}-\Delta, x_{\max}]$ belongs to the rightmost interval of X; and $[y_{\max}-\Delta, y_{\max}]$ belongs to the rightmost interval of Y, where $x_{\max}$ and $y_{\max}$ denotes the maximum of the support of X and Y. Define an event $\mathcal{M} : x_{\max}+y_{\max}-\delta \leq X+Y+\varepsilon \leq x_{\max}+y_{\max}+1$. If $\mathcal{M}$ occurs, note that ε is bounded by 1, we have

$$\mathbb{P}_X(X \leq x_{\max}-\delta|\mathcal{M}) = 0, \quad \mathbb{P}_Y(Y \leq y_{\max}-\delta|\mathcal{M}) = 0 \tag{32}$$

If $\delta < \Delta$, based on the definition of the interval, we have that

$$\mathbb{P}_X(X \leq x_{\max}-\delta) > 0, \quad \mathbb{P}_Y(Y \leq y_{\max}-\delta) > 0 \tag{33}$$

If $X+Y+\varepsilon \perp Y$ then $\mathbb{P}_Y(Y \leq y_{\max}-\delta) = \mathbb{P}_Y(Y \leq y_{\max}-\delta|\mathcal{M})$, which is not the case from the above equations (32) and (33). □

Proof of Theorem 3.1. The proof of the theorem resembles the proof of Theorem 4 in (Ahuja et al., 2021). Consider a solution to equation $\Phi^\dagger$,

$$\begin{aligned}\Phi^\dagger \cdot X^e &= \Phi^\dagger \cdot S(X^e_{\text{inv}}, X^e_{\text{spu}}) = \Phi_{\text{inv}} \cdot X^e_{\text{inv}} + \Phi_{\text{spu}} \cdot X^e_{\text{spu}} \\ &= \Big[\Phi_{\text{inv}} + \Phi_{\text{spu}} \cdot A\Big] \cdot X^e_{\text{inv}} + \Phi_{\text{spu}} \cdot W^e.\end{aligned} \tag{34}$$

and since $\Phi^\dagger$ achieves the error of q,

$$\mathrm{I}(w^+_{\text{inv}} \cdot X^e_{\text{inv}}) = \mathrm{I}(\Phi_{\text{inv}} \cdot X^e_{\text{inv}} + \Phi_{\text{spu}} \cdot X^e_{\text{spu}}) \tag{35}$$

In the next we prove Φ_{spu} by contradiction. Define $\Phi^+ = \Big(\Big[\Phi_{\text{inv}} + \Phi_{\text{spu}} \cdot A\Big], 0\Big)S^{-1}$. Observe that we can write $\Phi^\dagger \cdot X^e = \Phi^+ \cdot X^e + \Phi_{\text{spu}} \cdot W^e$. a) $\Phi_{\text{spu}} \cdot W^e \perp \Phi^+ \cdot X^e$ ($\Phi^+ \cdot X^e = \Big[\Phi_{\text{inv}} + \Phi_{\text{spu}} \cdot A\Big] \cdot X^e_{\text{inv}}$ and $X^e_{\text{inv}} \perp W^e$),

b.1) $\Phi^+ \cdot X^e$, $\Phi_{\text{spu}} \cdot W^e$ are discrete random variables with finite support of size at least two. (discrete case)

b.2) $\Phi^+ \cdot X^e$, $\Phi_{\text{spu}} \cdot W^e$ are continuous bounded random variables. (continuous case)

In the discrete case, from a), b.1), and Lemma A.1 it follows that

$$\lambda H(\Phi^+ \cdot X^e) + H(\Phi_{\text{spu}} \cdot W^e) > \lambda H(\Phi^\dagger \cdot X^e) \tag{36}$$

Rearranging the terms, we have

$$H(\Phi_{\text{spu}} \cdot W^e) - \lambda H(\Phi^\dagger \cdot X^e) > -\lambda H(\Phi^+ \cdot X^e) \tag{37}$$

Since $X^e_{\text{inv}} = G(X_{\text{inv}}, e)$, we have $H(\Phi^\dagger \cdot X^e|X_{\text{inv}}) = H(\Phi_{\text{spu}} \cdot W^e)$ and $H(\Phi^+ \cdot X^e|X_{\text{inv}}) = 0$. Hence, we get

$$H(\Phi^\dagger \cdot X^e|X_{\text{inv}}) - \lambda H(\Phi_{\text{spu}} \cdot W^e) > H(\Phi^+ \cdot X^e|X_{\text{inv}}) - H(\Phi^+ \cdot X^e). \tag{38}$$

and therefore, $\Phi^+ \cdot X^e$ would have a lower penalty. In the continuous case, the argument is similar by invoking a), b.2) and Lemma A.2. Φ^+ can achieve strictly lower penalty than $\Phi^\dagger$.

Following the proof of the first part of Theorem 4 in (Ahuja et al., 2021), we can show that Φ^+ achieves the same error of q in all the training environments. Thus Φ^+ is a strictly better solution $\Phi^\dagger$, which contradicts the optimality of $\Phi^\dagger$. Therefore, it follows that $\Phi_{\text{spu}} = 0$. And hence,

$$\mathrm{I}(w_{\text{inv}}^+ \cdot X_{\text{inv}}^e) = \mathrm{I}(\Phi_{\text{inv}} \cdot X_{\text{inv}}^e) \tag{39}$$

Based on Theorem 3 in (Ahuja et al., 2021), if a solution does not rely on spurious features and satisfies equation (39) for all the points in the support, then under the Assumption 3 such a solution solves the domain generalization problem (DG).

□