Deep Stable Learning for Out-Of-Distribution Generalization

Xingxuan Zhang, Peng Cui*, Renzhe Xu, Linjun Zhou, Yue He, Zheyan Shen
Department of Computer Science, Tsinghua University, Beijing, China
xingxuanzhang@hotmail.com, cuip@tsinghua.edu.cn, xrz199721@gmail.com,
{zhoulj16, heyue18, shenzy17}@mails.tsinghua.edu.cn

Abstract

Approaches based on deep neural networks have achieved striking performance when testing data and training data share similar distribution, but can significantly fail otherwise. Therefore, eliminating the impact of distribution shifts between training and testing data is crucial for building performance-promising deep models. Conventional methods assume either the known heterogeneity of training data (e.g. domain labels) or the approximately equal capacities of different domains. In this paper, we consider a more challenging case where neither of the above assumptions holds. We propose to address this problem by removing the dependencies between features via learning weights for training samples, which helps deep models get rid of spurious correlations and, in turn, concentrate more on the true connection between discriminative features and labels. Extensive experiments clearly demonstrate the effectiveness of our method on multiple distribution generalization benchmarks compared with state-of-the-art counterparts. Through extensive experiments on distribution generalization benchmarks including PACS, VLCS, MNIST-M, and NICO, we show the effectiveness of our method compared with state-of-the-art counterparts.

1. Introduction

Many machine learning approaches tend to exploit subtle statistical correlations existing in the training distribution for predictions which have been shown to be effective under the I.I.D. hypothesis, i.e., testing and training data is independently sampled from the identical distribution. In real cases, however, such a hypothesis can hardly be satisfied due to the complex generation mechanism of real data such as data selection biases, confounding factors, or other peculiarities [5, 62, 13, 55, 23]. The testing distribution may incur uncontrolled and unknown shifts from the training distribution, which makes most machine learning models fail to make trustworthy predictions [2, 59]. To address this issue, out-of-distribution (OOD) generalization is proposed for improving the generalization ability of models under distribution shifts [63, 32].

Figure 1: Visualization of saliency maps produced by the vanilla ResNet-18 model and StableNet when most of the training images containing dogs in the water. The lightness of the saliency map indicates how much attention that the models pay on particular area of the input image (i.e. lighter area plays a more crucial role for the prediction than the darker area). Due to the spurious correlation, the ResNet-18 model tends to focus on both dogs and the water while our model focuses mostly on dogs.

Essentially, when there incurs a distribution shift, the accuracy drop of current models is mainly caused by the spurious correlation between the irrelevant features (i.e. the features that are irrelevant to a given category, such as features of context, figure style, etc.) and category labels, and this kind of spurious correlations are intrinsically caused by the subtle correlations between irrelevant features and relevant features (i.e. the features that are relevant to a given category) [35, 45, 42, 2]. Taking the recognition task of 'dog' category as an example, as depicted in Figure 1, if dogs are

*Corresponing author, also with Beijing Key Lab of Networked Multimedia

in the water in most training images, the visual features of dogs and water would be strongly correlated, thus leading to the spurious correlation between visual features of water with the label 'dog'. As a result, when encountering images of dogs without water, or other objects (such as cats) with water, the model is prone to produce false predictions.

Recently, such distribution (domain) shift problems have been intensively studied in the *domain generalization (DG)* literature [48, 18, 29, 70, 36, 38]. The basic idea of DG is to divide a category into multiple domains so that irrelevant features vary across different domains while relevant features remain invariant [29, 39, 47]. Such training data makes it possible for a well-designed model to learn the invariant representations across domains and inhibit the negative effect from irrelevant features, leading to better generalization ability under distribution shifts. Some pioneering methods require clear and significant heterogeneity, namely that the domains are manually divided and labeled [69, 17, 54, 10, 49], which cannot be always satisfied in real applications. More recently, some methods are proposed to implicitly learn latent domains from data [52, 46, 68], but they implicitly assume that the latent domains are balanced, meaning that the training data is formed by balanced sampling from latent domains. In real cases, however, the assumption of domain balance can be easily violated, leading to the degeneration of these methods. This is also empirically validated in our experiments as shown in Section 4.

Here we consider a more realistic and challenging setting where the domains of training data are unknown and we do not implicitly assume that the latent domains are balanced. With this goal, a strand of research on stable learning are proposed [58, 33]. Given that the statistical dependence between relevant and irrelevant features is a major cause of model crash under distribution shifts, they propose to realize out-of-distribution generalization by decorrelating the relevant and irrelevant features. Since there is no extra supervision for separating relevant features from irrelevant features, a conservative solution is to decorrelate all features. Recently, this notion has been demonstrated to be effective in improving the generalization ability of linear models. [34] proposes a sample weighting approach with the goal of decorrelating input variables, and [59] theoretically proves why such sample weighting can make a linear model produce stable predictions under distribution shifts. But they are all developed under the constraints of linear frameworks. When extending these ideas into deep models to tackle more complicated data types like images, we confront two main challenges. First, the complex non-linear dependencies among features are much more difficult to be measured and eliminated than the linear ones. Second, the global sample weighting strategy in these methods requires excessive storage and computational cost in deep models, which is infeasible in practice.

To address these two challenges, we propose a method called **StableNet**. In terms of the first challenge, we propose a novel nonlinear feature decorrelation approach based on Random Fourier Features [53] with linear computational complexity. As for the second challenge, we propose an efficient optimization mechanism to perceive and remove correlations globally by iteratively saving and reloading features and weights of the model. These two modules are jointly optimized in our method. Moreover, as shown in Figure 1, StableNet can effectively partial out the irrelevant features (i.e. water) and leverage truly relevant features for prediction, leading to more stable performances in the wild non-stationary environments.

2. Related Works

Domain Generalization. Domain generalization (DG) considers the generalization capacities to unseen domains of deep models trained with multiple source domains. A common approach is to extract domain-invariant features over multiple source domains [18, 29, 37, 39, 47, 11, 25, 50, 56, 47] or to aggregate domain-specific modules [43, 44]. Several works propose to enlarge the available data space with augmentation of source domains [7, 57, 67, 52, 73, 72]. There are several approaches that exploit regularization with meta-learning [38, 11] and Invariant Risk Minimization (IRM) framework [2] for DG. Despite the promising results of DG methods in the well-designed experimental settings, some strong assumptions such as the manually divided and labeled domains and the balanced sampling process from each domain actually hinder the DG methods from real applications.

Feature Decorrelation. As the correlations between features affect or even impair the model prediction, several works have focused on remove such correlation in the training process. Some pioneering works based on Lasso framework [64, 8] propose to decorrelate features by adding a regularizer that imposes the highly correlated features not to be selected simultaneously. Recently, several works theoretically bridge the connections between correlation and model stability under misspecification [59, 34], and propose to address such a problem via a sample reweighting scheme. However, the above methods are all developed under linear frameworks which can not handle complex data types such as images and videos in computer vision applications. More related works and discussions are in Appendix A.

3. Sample Weighting for Distribution Generalization

We address the distribution shifts problem by weighting samples globally to directly decorrelate all the features for every input sample, thus statistical correlations between relevant and irrelevant features are eliminated. Concretely,

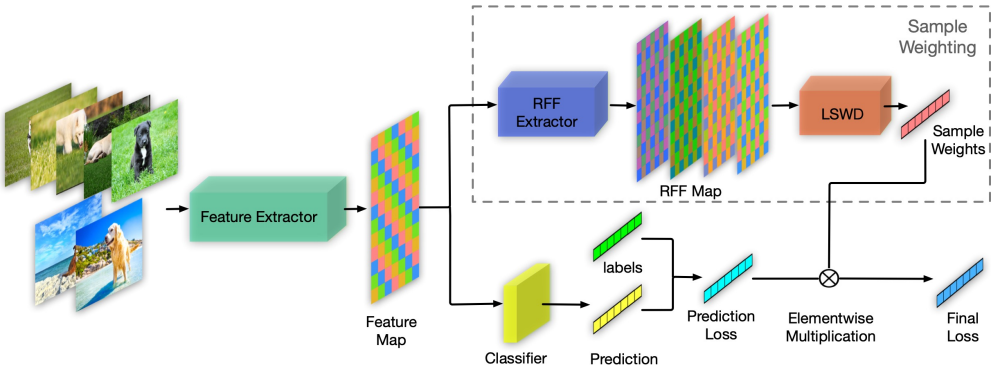


Figure 2: The overall architecture of the proposed StableNet. LSWD refers to *learning sample weighting for decorrelation* as described in Section 3.1. *Final loss* is used to optimized the classification network. Detailed learning procedure of StableNet is in Section 3.1 and Appendix B.1.

StableNet gets rid of both linear and non-linear dependencies between features by utilizing the characteristics of Random Fourier Features (RFF) and sample weighting. To adapt the global decorrelation method to modern deep models, we further propose the saving and reloading global correlation mechanism, to decrease the usage of storage and computational cost when the training data are of a large scale. The formulations and theoretical explanations are shown in Section 3.1. In Section 3.2, we introduce the saving and reloading global correlation method, which makes calculating correlation globally possible with deep models. **Notations** $\mathcal{X} \subset \mathbb{R}^{m_X}$ denotes the space of raw pixels, $\mathcal{Y} \subset \mathbb{R}^{m_Y}$ denotes the outcome space and $\mathcal{Z} \subset \mathbb{R}^{m_Z}$ denotes the representation space. m_X, m_Y, m_Z are the dimensions of space $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, respectively. $f : \mathcal{X} \to \mathcal{Z}$ denotes the representation function and $g : \mathcal{Z} \to \mathcal{Y}$ denotes the prediction function. We have n samples $\mathbf{X} \subset \mathbb{R}^{n \times m_X}$ with labels $\mathbf{Y} \subset \mathbb{R}^{n \times m_Y}$ and we use $\mathbf{X}_i$ and y_i to denote the i-th sample. The representations learned by neural networks are donated as $\mathbf{Z} \subset \mathbb{R}^{n \times m_Z}$ and the i-th variable in the representation space is donated as $\mathbf{Z}_{:,i}$. We use $\mathbf{w} \in \mathbb{R}^n$ to denote sample weights. $\mathbf{u}$ and $\mathbf{v}$ are Random Fourier Features mapping functions.

3.1. Sample weighting with RFF

Independence testing statistics To eliminate the dependence between any pair of features $\mathbf{Z}_{:,i}$ and $\mathbf{Z}_{:,j}$ in the representation space, we introduce hypothesis testing statistics that measures the independence between random variables. Suppose there are two one-dimensional random variables A, B (Here we use A and B to represent random variables instead of $\mathbf{Z}_{:,i}$ and $\mathbf{Z}_{:,j}$ for simplicity of notation.) and we sample $(A_1, A_2, \dots A_n)$ and $(B_1, B_2, \dots B_n)$ from the distribution of A and B, respectively. The main problem is how relevant these two variables are based on the samples.

Consider a measurable, positive definite kernel k_A on the domain of random variable A and the corresponding RKHS is denoted by $\mathcal{H}_A$. If k_B and $\mathcal{H}_B$ are similarly defined, the cross-covariance operator Σ_{AB} [14] from $\mathcal{H}_B$ to $\mathcal{H}_A$ is as follows:

$$\begin{aligned} &\langle h_A, \Sigma_{AB} h_B \rangle \\ =&\mathbb{E}_{AB}\left[h_A(A)h_B(B)\right] - \mathbb{E}_A[h_A(A)]\mathbb{E}_B[h_B(B)] \end{aligned} \tag{1}$$

for all $h_A \in \mathcal{H}_A$ and $h_B \in \mathcal{H}_B$. Then, the independence can be determined by the following proposition [15].

Proposition 3.1 *If the product $k_A k_B$ is characteristic, $\mathbb{E}[k_A(A, A)] < \infty$ and $\mathbb{E}[k_B(B, B)] < \infty$, we have*

$$\Sigma_{AB} = 0 \Longleftrightarrow A \perp B \tag{2}$$

Hilbert-Schmidt Independence Criterion (HSIC) [20], which requires that the squared Hilbert-Schmidt norm of Σ_{AB} should be zero, can be applied as a criterion to supervise feature decorrelation [3]. However, the calculation of HSIC requires noticeable computational cost which grows as the batch size of training data increases, so it is inapplicable to training deep models on large datasets. More approaches of independence test are discussed in Appendix B.2. Actually, Frobenius norm corresponds to the Hilbert-Schmidt norm in Euclidean space [61], so that the independent testing statistic can be based on Frobenius norm.

Let the partial cross-covariance matrix be:

$$\begin{aligned} \hat{\Sigma}_{AB} = \frac{1}{n-1} \sum_{i=1}^{n} \Bigg[\left(\mathbf{u}(A_i) - \frac{1}{n} \sum_{j=1}^{n} \mathbf{u}(A_j) \right)^T \cdot \\ \left(\mathbf{v}(B_i) - \frac{1}{n} \sum_{j=1}^{n} \mathbf{v}(B_j) \right) \Bigg], \end{aligned} \tag{3}$$

where

$$\begin{aligned} \mathbf{u}(A) &= \left(u_1(A), u_2(A), \dots u_{n_A}(A)\right), u_j(A) \in \mathcal{H}_{\text{RFF}}, \forall j, \\ \mathbf{v}(B) &= \left(v_1(B), v_2(B), \dots v_{n_B}(B)\right), v_j(B) \in \mathcal{H}_{\text{RFF}}, \forall j. \end{aligned} \tag{4}$$

Here we sample n_A and n_B functions from $\mathcal{H}_{\text{RFF}}$ respectively and $\mathcal{H}_{\text{RFF}}$ denotes the function space of Random Fourier Features with the following form

$$\mathcal{H}_{\text{RFF}} = \big\{h : x \to \sqrt{2}\cos(\omega x + \phi) \mid \omega \sim N(0,1), \phi \sim \text{Uniform}(0, 2\pi)\big\}, \tag{5}$$

i.e. ω is sampled from the standard Normal distribution and ϕ is sampled from the Uniform distribution. Then, the independence testing statistic I_{AB} is defined as the Frobenius norm of the partial cross-covariance matrix, *i.e.*, $I_{AB} = \left\|\hat{\Sigma}_{AB}\right\|_F^2$.

Notice that I_{AB} is always non-negative. As I_{AB} decreases to zero, the two variables A and B tends to be independent. Thus I_{AB} can effectively measure the independence between random variables. The accuracy of independence test grows as n_A and n_B increase. Empirically, setting both n_A and n_B to 5 is solid enough to judge the independence of random variables [61].

Learning sample weights for decorrelation Inspired by [34], we propose to eliminate the dependence between features in the representation space via sample weighting and measure general independence via RFF.

We use $\mathbf{w} \in \mathbb{R}_+^n$ to denote the sample weights and $\sum_{i=1}^n w_i = n$. After weighting, the partial cross-covariance matrix for random variables A and B in Equation 3 can be calculated as follows:

$$\hat{\Sigma}_{AB;\mathbf{w}} = \frac{1}{n-1}\sum_{i=1}^{n}\Bigg[\bigg(w_i\mathbf{u}(A_i) - \frac{1}{n}\sum_{j=1}^{n} w_j\mathbf{u}(A_j)\bigg)^T \cdot \bigg(w_i\mathbf{v}(B_i) - \frac{1}{n}\sum_{j=1}^{n} w_j\mathbf{v}(B_j)\bigg)\Bigg]. \tag{6}$$

Here $\mathbf{u}$ and $\mathbf{v}$ are the RFF mapping functions explained in Equation 4. StableNet targets independence between any pair of features. Specifically, for feature $\mathbf{Z}_{:,i}$ and $\mathbf{Z}_{:,j}$, the corresponding partial cross-covariance matrix should be $\left\|\hat{\Sigma}_{Z_{:,i}Z_{:,j};\mathbf{w}}\right\|_F^2$, shown in Equation 6. We propose to optimize $\mathbf{w}$ by

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\Delta_n} \sum_{1\le i<j\le m_Z} \left\|\hat{\Sigma}_{\mathbf{Z}_{:,i}\mathbf{Z}_{:,j};\mathbf{w}}\right\|_F^2, \tag{7}$$

where $\Delta_n = \big\{\mathbf{w} \in \mathbb{R}_+^n \mid \sum_{i=1}^n w_i = n\big\}$. Hence, weighting training samples with the optimal $\mathbf{w}^*$ can mitigate the dependence between features to the greatest extent

Generally, our algorithm iteratively optimize sample weights $\mathbf{w}$, representation function f, and prediction function g as follows:

$$\begin{aligned} f^{(t+1)}, g^{(t+1)} =& \arg\min_{f,g} \sum_{i=1}^{n} w_i^{(t)} L(g(f(\mathbf{X}_i)), y_i), \\ \mathbf{w}^{(t+1)} =& \arg\min_{\mathbf{w}\in\Delta_n} \sum_{1\le i<j\le m_Z} \left\|\hat{\Sigma}_{\mathbf{Z}_{:,i}^{(t+1)}\mathbf{Z}_{:,j}^{(t+1)};\mathbf{w}}\right\|_F^2. \end{aligned} \tag{8}$$

where $\mathbf{Z}^{(t+1)} = f^{(t+1)}(\mathbf{X})$, $L(\cdot,\cdot)$ represents the cross entropy loss function and t represents the time stamp. Initially, $\mathbf{w}^{(0)} = (1, 1, \ldots, 1)^T$.

3.2. Learning sample weights globally

Equation 8 requires a specific weight learned for each sample. However, in practice, especially for deep learning tasks, it requires enormous storage and computational cost to learn sample weights globally. Moreover, with SGD for optimization, only part of the samples are observed in each batch, hence global weights for all samples cannot be learned. In this part, we propose a saving and reloading method, which merges and saves features and sample weights encountered in the training phase and reloads them as global knowledge of all the training data to optimize sample weights.

For each batch, the features used to optimize the sample weights are generated as follows:

$$\begin{aligned} \mathbf{Z}_O &= \text{Concat}\left(\mathbf{Z}_{G1}, \mathbf{Z}_{G2}, \cdots, \mathbf{Z}_{Gk}, \mathbf{Z}_L\right), \\ \mathbf{w}_O &= \text{Concat}\left(\mathbf{w}_{G1}, \mathbf{w}_{G2}, \cdots, \mathbf{w}_{Gk}, \mathbf{w}_L\right). \end{aligned} \tag{9}$$

Here we slightly abuse the notation $\mathbf{Z}_O$ and $\mathbf{w}_O$ to mean the features and weights used to optimize the new sample weights, respectively, $\mathbf{Z}_{G1}, \cdots, \mathbf{Z}_{Gk}$, $\mathbf{w}_{G1}, \cdots, \mathbf{w}_{Gk}$ are global features and weights, which are updated at the end of each batch and represent global information of the whole training dataset. $\mathbf{Z}_L$ and $\mathbf{w}_L$ are features and weights in the current batch, representing the local information. The operation for merging all features in Equation 9 is the concatenating operation along samples, *i.e.* if the batch size is B, $\mathbf{Z}_O$ is a matrix of size $((k+1)B) \times m_Z$ and $\mathbf{w}_O$ is a $((k+1)B)$-dimensional vector. In this way, we reduce the storage and the computational cost from $O(N)$ to $O(kB)$. While training for each batch, we keep $\mathbf{w}_{Gi}$ fixed and only $\mathbf{w}_L$ is learnable under Equation 8. At the end of each iteration of training, we fuse the global information $(\mathbf{Z}_{Gi}, \mathbf{w}_{Gi})$ and the local information $(\mathbf{Z}_L, \mathbf{w}_L)$ as follows:

$$\begin{aligned} \mathbf{Z}'_{Gi} &= \alpha_i\mathbf{Z}_{Gi} + (1-\alpha_i)\mathbf{Z}_L, \\ \mathbf{w}'_{Gi} &= \alpha_i\mathbf{w}_{Gi} + (1-\alpha_i)\mathbf{w}_L. \end{aligned} \tag{10}$$

Here for each group of global information $(\mathbf{Z}_{Gi}, \mathbf{w}_{Gi})$, we use k different smoothing parameters α_i for considering both long-term memory (α_i is large) and short-term memory (α_i is small) in global information and k indicates that

the presaved features are k times of that of original features. Finally, we substitute all $(\mathbf{Z}_{Gi}, \mathbf{w}_{Gi})$ with $(\mathbf{Z}'_{Gi}, \mathbf{w}'_{Gi})$ for the next batch.

In the training phase, we iteratively optimize sample weights and model parameters with Equation 8. In the inference phase, the predictive model directly conduct prediction without any calculation of sample weights. The detailed procedure of our method is shown in Appendix B.1.

4. Experiments

4.1. Experimental settings and datasets

We validate StableNet in a variety of settings. To cover more general and challenging cases of distribution shifts, we adopt four experimental settings as follows:

Unbalanced. In the common DG setting, the capacities of source domains are assumed to be comparable. However, considering most datasets are a mixture of latent unknown domains, one can hardly assume that the amount of samples from these domains are consistent since these datasets are not generated by equally sampling from latent domains. We simulate this scenario with this setting. Domains are split into source domains and target domains. The capacities of various domains can vary significantly. Note that this setting, where the capacities of available domains are unbalanced while the proportion of each class remains consistent across domains, is completely different from the settings of the class imbalance problem. This setting is to evaluate the generalization ability of models when the heterogeneity is unclear and insignificant.

Flexible. We consider a more challenging but common in real-world setting where domains for different categories can be various. For instance, birds can be on trees but hardly in the water while fishes are the opposite. If we consider the backgrounds in images as an indicator of domain division, images for class 'bird' can be divided into domain 'on tree' but cannot into domain 'in water' while images for class 'fish' are otherwise, resulting in the diversity of domains among different classes. Thus this setting simulates a widely existing scenario in the real-world. In such cases, the level of the distribution shifts varies in different classes, requiring a strong ability of generalization given the statistical correlations between relevant features and category-irrelevant features vary.

Adversarial. We consider the most challenging scenario, where the model is under adversarial attack and the spurious correlations between domains and labels are strong and misleading. For instance, we assume a scenario where the category 'dog' is usually associated with the domain 'grass' and the category 'cat' with the domain 'sofa' in the training data, while the category 'dog' is usually associated with the domain 'sofa' and the category 'cat' with the domain 'grass' in the testing data. If the ratio of domain 'grass' in the images from class 'dog' is significantly higher than others, the predictive model may tend to recognize grass as a dog.

Classic. This setting is the same as the common setting in DG. The capacities of various domains are comparable. Therefore this setting is to evaluate the generalization ability of models when the heterogeneity of training data is significant and clear, which is less challenging compared with the previous three settings.

Datasets. We consider four datasets to carry through these four settings, namely PACS [36], VLCS [66], MNIST-M [16] and NICO [22]. Introduction to these datasets and details of implementation are in Appendix C.1.

4.2. Unbalanced setting

Given this setting requires all the classes in the dataset share the same candidate set of domains, which is incompatible with NICO, we adopt PACS and VLCS for this setting. Three domains are considered as source domains and the other one as target. To make the amount of data from heterogeneous sources clearly differentiated, we set one domain as the dominant domain. For each target domain, we randomly select one domain from the source domains as the dominant source domain and adjust the ratio of data from the dominant domain and the other two domains. Details of ratios and partition are shown in Appendix C.2.

Here we show the results when the capacity ratio of three source domains is 5:1:1 in Table 1 and our method outperforms other methods in all the target domains on both PACS and VLCS. Moreover, StableNet achieves best performance consistently under all the other ratios as shown in Appendix C.2. These results indicate that the subtle statistical correlations between relevant and irrelevant features are strong enough to significantly harm the generalization across domains. When the correlations are eliminated, the model is able to learn the true connections between relevant features and labels and inference according to them only, thus generalize better. For adversarially trained methods like DG-MMLD [46], the supervision from minor domains is insufficient and the ability of the model to discriminate irrelevant features is impaired. For augmentation of source domains based methods like M-ADA [52], the impact of the dominant domain is not diminished while the minor ones are still insignificant after the augmentation. Methods like RSC [27] adopt regularization to prevent the model from overfitting on source domains and the samples from minor domains can be considered as outliers and ignored. Therefore, the subtle correlations between relevant features and irrelevant features especially in minor domains are not eliminated.

4.3. Unbalanced + flexible setting

We adopt PACS, VLCS and NICO to evaluate the *unbalanced + flexible* setting. For PACS and VLCS, we randomly select one domain as the dominant domain for each

Table 1: Results of the *unbalanced* setting on PACS and VLCS. We reimplement the methods that require no domain labels on PACS and VLCS with ResNet18 [21] which is pretrained on ImageNet [9] as the backbone network for all the methods. The reported results are average over three repetitions of each run. The title of each column indicates the name of the domain used as target. The best results of all methods are highlighted with the bold font and the second with underscore.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen [7]	72.76	69.21	64.90	91.24	74.53	85.20	59.73	62.64	<u>50.59</u>	<u>64.54</u>
M-ADA [52]	61.53	68.76	58.49	83.21	68.00	70.29	55.44	49.96	37.78	53.37
DG-MMLD [46]	64.25	<u>70.31</u>	64.16	91.64	72.59	79.76	57.93	65.25	44.61	61.89
RSC [27]	<u>75.72</u>	68.50	<u>66.10</u>	<u>93.93</u>	<u>76.06</u>	83.82	59.92	<u>64.49</u>	49.08	64.33
ResNet-18	68.41	67.32	65.75	90.22	72.93	<u>80.02</u>	<u>60.21</u>	58.33	47.59	61.54
StableNet (ours)	**80.16**	**74.15**	**70.10**	**94.24**	**79.66**	**88.25**	**62.59**	**65.77**	**55.34**	**67.99**

Table 2: Results of the *unbalanced + flexible* setting on PACS, VLCS and NICO. For details about the number of runs, meaning of column titles and fonts, see Table 1.

	JiGen	M-ADA	DG-MMLD	RSC	ResNet-18	StableNet (ours)
PACS	40.31	30.32	<u>42.65</u>	39.49	39.02	**45.14**
VLCS	76.75	69.58	<u>78.96</u>	74.81	73.77	**79.15**
NICO	54.42	40.78	47.18	<u>57.59</u>	51.71	**59.76**

class, and another domain as the target. For NICO, there are 10 domains for each class, 8 out of which are selected as the source and 2 as the target. We adjust the ratio of the dominant domain to minor domains to adjust the level of distribution shifts. Here we report the results when the dominant ratio is 5:1:1. Details and more results of other divisions are shown in Appendix C.3.

The results are shown in Table 2. M-ADA and DG-MMLD fail to outperform ResNet-18 on NICO under this setting. M-ADA, which generates images for training with an autoencoder, may fail when the training data are large-scale real-world images and the distribution shifts are not caused by random disturbance. DG-MMLD generates domain labels with clustering and may fail when the data lack explicit heterogeneity or the number of latent domains is too large for clustering. In contrast, StableNet shows a strong ability of generalization when the input data are with complicated structure especially real-world images from unlimited resources. StableNet can capture various forms of dependencies and balance the distribution of input data. On PACS and VLCS, StableNet also outperforms state-of-the-art methods, showing the effectiveness of removing statistical dependencies between features especially when the source domains for different categories are not consistent. More experimental results are in Appendix C.3.

4.4. Unbalanced + flexible + adversarial setting

To exploit the effect of various levels of adversarial attack, we adopt MNIST-M to evaluate our method owing to the numerous (200) optional domains in MNIST-M. Domains in PACS and VLCS are insufficient to generate multiple adversarial levels. Hence, we generate a new MNIST-M dataset with three rules: 1) for a given category, there is no overlap between the domains in training and testing; 2) a background image is randomly chosen for each category in the training set, and contexts cropped in the same image are assigned as dominant contexts (domains) for another category in test data so that there are strong spurious correlations between labels and domains; 3) the ratio of dominant context to other contexts varies from 9.5:1 to 1:1 to generate settings with different levels of distribution shifts. Detailed data generating method, adopted backbone network and sample images are in Appendix C.4.

The results are shown in Table 3. As the dominant ratio increases, the spurious correlation between domains and categories becomes stronger so that the performance of predictive models drops. When the imbalance in visual features is significant, our method achieves noticeable improvement compared with baseline methods. For regularization-based methods such as RSC, they tend to weaken the supervision from minor domains which may be considered as outliers and therefore the spurious correlations between irrelevant features and labels are strengthened under adversarial attacks, resulting in even poorer results compared with the vanilla ResNet model. As shown in Table 3, RSC fails to outperform vanilla CNNs.

4.5. Classic setting

The *classic* setting is the same as the common setting in DG. Domains are split into source domains and target domains. The capacities of various domains are comparable. Given this setting requires all the classes in the dataset to

Table 3: Results of the *unbalanced + flexible + adversarial* setting on MNIST-M. Random donates each digit is blended over a randomly chosen background. DR0.5 donates that in each class, the proportion of the dominant domain in all the training data is 50% and other notations with 'DR' are similar.

Settings	Random	DR0.5	DR0.6	DR0.7	DR0.8	DR0.9	DR0.95	Avg.
JiGen	97.18	94.97	92.99	90.64	78.97	68.79	69.34	84.70
M-ADA	95.92	94.45	92.29	88.87	85.89	70.32	67.08	84.97
DG-MMLD	96.89	94.61	92.59	89.72	**88.44**	69.13	71.39	86.11
RSC	96.94	93.43	89.44	85.78	81.68	69.15	65.12	83.08
CNNs	96.93	93.76	91.93	88.13	81.48	68.43	66.11	83.82
StableNet (ours)	**97.35**	**95.33**	**93.49**	**91.24**	87.04	**75.69**	**75.46**	**87.94**

Table 4: Results of the *classic* setting on PACS and VLCS. All the results on PACS are obtained from the original papers of these methods. We reimplement the methods that require no domain labels on VLCS since these methods are tested with AlexNet [31] in original papers while we adopt ResNet18 [21] as the backbone network for all the methods. The methods that require domain labels are labelled with asterisk.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen	79.42	75.25	71.35	96.03	80.51	96.17	62.06	70.93	71.40	75.14
M-ADA	64.29	72.91	67.21	88.23	73.16	74.33	48.38	45.31	33.82	50.46
DG-MMLD	81.28	77.16	72.29	96.09	81.83	**97.01**	62.20	73.01	72.49	76.18
D-SAM* [12]	77.33	72.43	77.83	95.30	80.72	-	-	-	-	-
Epi-FCR* [38]	82.10	77.00	73.00	93.90	81.50	-	-	-	-	-
FAR* [28]	79.30	77.70	74.70	95.30	81.70	-	-	-	-	-
MetaReg* [4]	**83.70**	77.20	70.30	95.50	81.70	-	-	-	-	-
RSC	83.43	**80.31**	**80.85**	95.99	**85.15**	96.21	62.51	**73.81**	72.10	76.16
ResNet-18	76.61	73.60	76.08	93.31	79.90	91.86	61.81	67.48	68.77	72.48
StableNet (ours)	81.74	79.91	80.50	**96.53**	84.69	96.67	**65.36**	73.59	**74.97**	**77.65**

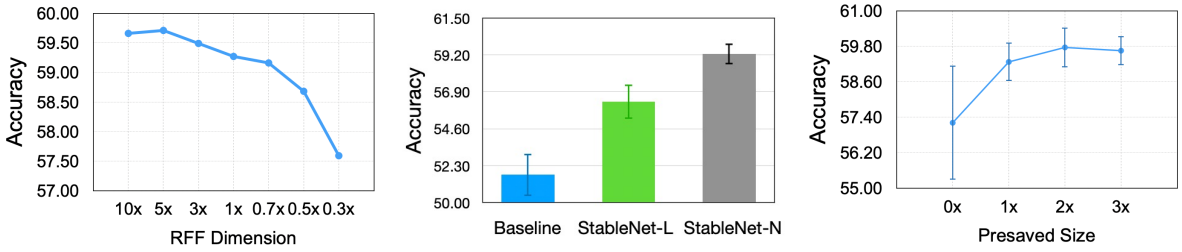
share the same candidate set of domains, which is incompatible with NICO, we adopt PACS and VLCS for this setting. We follow the experimental protocol of [7, 46] for both the datasets and utilize three domains as source domains and the remaining one as the target.

The results are shown in Table 4. On VLCS, StableNet outperforms other state-of-the-art methods in two out of four target cases and achieves the highest average accuracy. On PACS, StableNet achieves the highest accuracy on the target domain 'photo' and comparable average accuracy (0.46% less) compared with the state-of-the-art method, RSC. The accuracy gap between StableNet and baseline indicates that even when the numbers of samples from different source domains are approximately the same, the subtle statistical correlations between relevant features and irrelevant features still hold strong and the model generalizes across domains better when the correlations are eliminated.

4.6. Ablation study

StableNet relies on Random Fourier Features sampled from Gaussian to balance the training data. The more features are sampled, the more independent the final representations are. In practice, however, generating more features requires more computational cost. In this ablation study, we exploit the effect of sampling size for Random Fourier Features. Moreover, inspired by [65], one can further reduce the feature dimension by randomly selecting features used to calculate dependence with different ratios. Figure 3 shows the results of StableNet with different dimensions of Random Fourier Features. If we remove all the Random Fourier Features, our regularizer in Equation 12 degenerates and can only model the linear correlation between features. Figure 2(a) demonstrates the effectiveness of eliminating non-linear dependence between representations. From Figure 2(b), the non-linear dependence is common in vision features and keep deep models from learning true dependence between input images and category labels.

We further exploit the effect of the size of presaved features and weights in Equation 9 and the results are shown in Figure 2(c). When the size of presaved features is reduced to 0, sample weights are learned inside of each batch, yielding noticeable variance. Generally, as the presaving size increases, the accuracy raises slightly and the variance drops significantly, indicating that presaved features help to learn sample weights globally and therefore the generalization ability of the model is more stable.



(a) (b) (c)

Figure 3: Results of ablation study on NICO. All the experiments adopt NICO since NICO consists of a wide range of domains and objects and all domains come from real-world images which make the indication of results more reliable. The RFF dimension in (a) indicates the dimension of Fourier features, where *10x* indicates that the dimension of Fourier features are 10 times the size of original features and *0.3x* indicates the sampling ratio is 30%. *StableNet-N* and *StableNet-L* indicate the original StableNet and the degenerated version of StableNet that only eliminates the linear correlation between features. *Presaved size* in (c) indicates the dimension of the presaved features and *0x* indicates no features are saved.

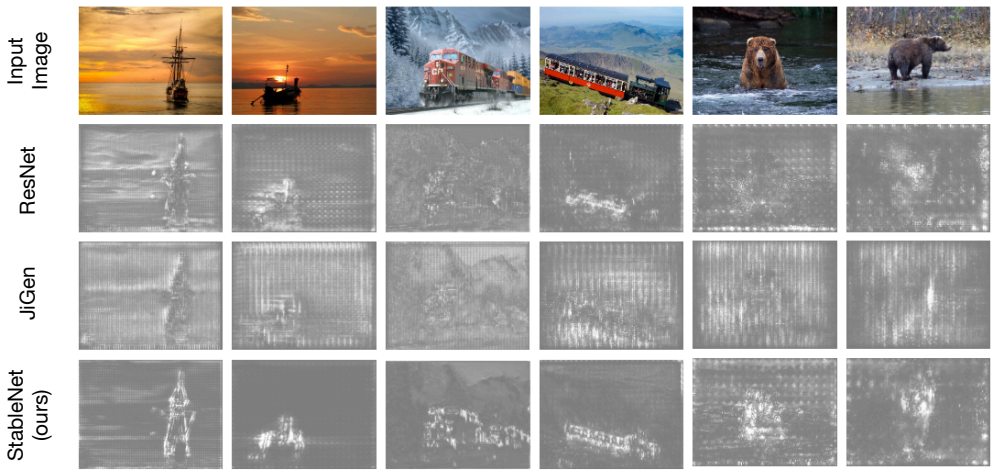


Figure 4: Saliency maps of the ResNet-18 model and StableNet. The brighter the pixel is, the more contributions it makes to prediction.

4.7. Saliency map

An intuitive type of explanation for image classification models is to identify pixels that have a strong influence on the final decision [60]. To demonstrate whether the model focuses on the object or the context (domain) while conducting prediction, we visualize the gradient of the class score function with respect to the input pixels. In the case of stable learning, we adopt the same backbone architecture for all methods, so that we adopt *smoothed* gradient as suggested by [1], which generates saliency maps depending on the learned parameters of the models instead of the architecture. Visualization results are shown in Figure 4. Saliency maps of the baseline model show that various contexts draw noticeable focus of the classifier while fail to make decisive contributions to our model. More visualization results are in Appendix C.6, which further demonstrate that StableNet focuses more on visual parts which are both distinguishing and invariant when the postures or positions of objects vary.

5. Conclusion

In the paper, to improve the generalization of deep models under distribution shifts, we proposed a novel method called StableNet which can eliminate the statistical correlation between relevant and irrelevant features via sample weighting. Extensive experiments across a wide range of settings demonstrated the effectiveness of our method.

Acknowledgement

This work was supported in part by National Key R&D Program of China (No. 2018AAA0102004, No. 2020AAA0106300), National Natural Science Foundation of China (No. U1936219, 61521002, 61772304), Beijing Academy of Artificial Intelligence (BAAI), and a grant from the Institute for Guo Qiang, Tsinghua University.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018. 8

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

[3] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. *arXiv preprint arXiv:1910.02806*, 2019. 3, 12

[4] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 998–1008. Curran Associates, Inc., 2018. 7

[5] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019. 1

[6] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018. 12

[7] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 2, 6, 7, 13, 14

[8] Sibao Chen, Chris HQ Ding, Bin Luo, and Ying Xie. Uncorrelated lasso. In *AAAI*, 2013. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 13

[10] Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing*, 27(1):304–313, 2017. 2

[11] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32:6450–6461, 2019. 2

[12] Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, pages 187–198. Springer, 2018. 7

[13] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019. 1

[14] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004. 3

[15] Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in neural information processing systems*, pages 489–496, 2008. 3

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 5

[17] 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. 2

[18] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015. 2, 13

[19] 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. 12

[20] Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592, 2008. 3

[21] 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*, pages 770–778, 2016. 6, 7, 12, 13

[22] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, page 107383, 2020. 5

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[24] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018. 12

[25] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis. In *Uncertainty in Artificial Intelligence*, pages 292–302. PMLR, 2020. 2

[26] Wen-Chin Huang, Hao Luo, Hsin-Te Hwang, Chen-Chou Lo, Yu-Huai Peng, Yu Tsao, and Hsin-Min Wang. Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020. 12

[27] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. *arXiv preprint arXiv:2007.02454*, 5(6), 2020. 5, 6, 13, 14

[28] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*, 2020. 7

[29] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 2

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 12

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 7

[32] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020. 1

[33] K. Kuang, R. Xiong, P. Cui, S. Athey, and B. Li. Stable prediction across unknown environments. *Research Papers*, 2018. 2

[34] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. Stable prediction with model misspecification and agnostic distribution shift. In *AAAI*, pages 4485–4492, 2020. 2, 4

[35] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1

[36] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 2, 5, 13

[37] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017. 2

[38] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019. 2, 7

[39] 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*, pages 5400–5409, 2018. 2

[40] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020. 12

[41] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 12, 14

[42] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6979–6987, 2017. 1

[43] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1353–1357. IEEE, 2018. 2

[44] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Robust place categorization with deep domain generalization. *IEEE Robotics and Automation Letters*, 3(3):2093–2100, 2018. 2

[45] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018. 1

[46] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, pages 11749–11756, 2020. 2, 5, 6, 7, 13, 14

[47] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017. 2

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

[49] Li Niu, Wen Li, and Dong Xu. Multi-view domain generalization for visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4201, 2015. 2

[50] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, pages 7728–7738. PMLR, 2020. 2

[51] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019. 12

[52] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 2, 5, 6, 13, 14

[53] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008. 2

[54] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pages 3236–3246, 2017. 2

[55] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 1

[56] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275*, 3(6):7, 2019. 2

[57] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018. 2

[58] Z. Shen, P. Cui, K. Kuang, B. Li, and P. Chen. On image classification: Correlation v.s. causality. 2017. 2

[59] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. In *AAAI*, pages 5692–5699, 2020. 1, 2

[60] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 8

[61] Eric V Strobl, Kun Zhang, and Shyam Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 7(1), 2019. 3, 4

[62] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 1

[63] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. 2019. 1

[64] Masaaki Takada, Taiji Suzuki, and Hironori Fujisawa. Independently interpretable lasso: A new regularizer for sparse regression with uncorrelated variables. In *International Conference on Artificial Intelligence and Statistics*, pages 454–463, 2018. 2

[65] Antti J Tanskanen, Jani Lukkarinen, and Kari Vatanen. Random selection of factors preserves the correlation structure in a linear factor model to a high degree. *Plos one*, 13(12):e0206551, 2018. 7

[66] A Torralba and AA Efros. Unbiased look at dataset bias. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. 5

[67] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, pages 5334–5344, 2018. 2

[68] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019. 2

[69] Haohan Wang, Aaksha Meghawat, Louis-Philippe Morency, and Eric P Xing. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 949–954. IEEE, 2017. 2

[70] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. *arXiv preprint arXiv:2007.09316*, 2020. 2

[71] Xinyue Wang, Yilin Lyu, and Liping Jing. Deep generative model for robust imbalance classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14124–14133, 2020. 12

[72] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032, 2020. 2

[73] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020. 2

A. Method

A.1. Detailed Training Procedure of StableNet

In the training phase, StableNet learns a set of sample weights for each batch with the global knowledge of correlations between features saved before. The parameters of the model and the sample weights are optimized iteratively.

As shown in Algorithm 1, for each input batch $(\mathbf{X}_L, \mathbf{Y}_L)$, the corresponding representations $\mathbf{Z}_L = f(\mathbf{X}_L)$ is concatenated with pre-saved global representations $\mathbf{Z}_{G1}, \mathbf{Z}_{G2}, \cdots, \mathbf{Z}_{Gk}$, as shown in Equation 11.

$$\mathbf{Z}_O = \text{Concat}\left(\mathbf{Z}_{G1}, \mathbf{Z}_{G2}, \cdots, \mathbf{Z}_{Gk}, \mathbf{Z}_L\right). \tag{11}$$

Then the model learns the local weights with global weights $\mathbf{w}_{G1}, \mathbf{w}_{G2}, \cdots, \mathbf{w}_{Gk}$ by optimizing Equation 12.

$$\mathbf{w}_L = \underset{\mathbf{w}\in\Delta_B}{\arg\min} \sum_{1\le i<j\le m_Z} \left\| \hat{\Sigma}_{\mathbf{Z}_{O_{:,i}}\mathbf{Z}_{O_{:,j}};\mathbf{w}_O} \right\|_F^2, \tag{12}$$

where B is the batch size and

$$\mathbf{w}_O = \text{Concat}\left(\mathbf{w}_{G1}, \mathbf{w}_{G2}, \cdots, \mathbf{w}_{Gk}, \mathbf{w}\right).$$

The loss for optimizing the representation extractor f and the classifier g is calculated by conducting sample weights $\mathbf{w}_L$ and penalties for input samples as shown in Equation 13.

$$L_{f,g} = \sum_{i=1}^{B} \mathbf{w}_{L_i} L(g(f(\mathbf{X}_{L_i})), \mathbf{Y}_{L_i}). \tag{13}$$

The present features and weights are integrated with the previous global features and weights as shown in Equation 14.

$$\begin{aligned} \mathbf{Z}'_{Gi} &= \alpha_i \mathbf{Z}_{Gi} + (1-\alpha_i)\mathbf{Z}_L, \\ \mathbf{w}'_{Gi} &= \alpha_i \mathbf{Z}_{Gi} + (1-\alpha_i)\mathbf{w}_L. \end{aligned} \tag{14}$$

In the inference phase, given the back propagation is disabled, StableNet escapes the sample weighting phase and conduct prediction directly.

In practice, the optimization also requires a regularizer of weight decay. We set the weight of the regularizer to 0.3 and the learning rate for sample weights to 3.0 unless otherwise noted.

A.2. Other Approaches for Independence Test

Despite the fact that there are several approaches to be used as the supervision of feature independence, they can hardly be used to optimize deep models within acceptable cost. For instance, Hilbert-Schmidt Independence Criterion (HSIC), which can be used in independent component analysis (ICA) [19], is applied as a criterion for feature decorrelation in [3]. However, the calculation of HSIC requires noticeable computational cost which grows as the batch size of training data increases, so it is inapplicable to training deep models on large datasets. Another approach, Mutual Information (MI), requires variational bounds to estimate [51], which are hard to be assembled with common convolutional networks such as ResNet [21].

A.3. Correlation between the Unbalanced Setting and the Problem of Unbalanced Classes

There are many works focus on the class imbalance problem[40, 41, 71]. But our *unbalanced* setting is a completely different problem. The key of problem of class imbalance is learning a better classifier to improve the recognition accuracy of minor classes, while the *unbalanced* setting valuates the ability of learning the correlation between object-relevant features and labels using the heterogeneity of unbalanced domains. In our setting, the proportion of different classes , which our settings do not focus on, may be different in both the *unbalanced* setting and the *classic* setting. Experiments show that methods for the class imbalance problem is not effective (comparable or worse compared with ResNet-18) in the *unbalanced* setting.

A.4. Correlation between our method and feature disentanglement

There are many works attempt learning disentangled features for robust or explainable representations[26, 6, 24] with strong constrains on features. These disentanglement methods such as VAE [30] force features to be disentangled, which changes the semantic implication of features, while StableNet learns sample weights to adjust the data structure while the semantic of features is not effected given disentanglement is not our target but a path to learn true correlation between relevant features and labels. We fail to train VAE-based models that outperform the ResNet-18 model in all of our settings.

Algorithm 1 *Training procedure of StableNet*

Input: EPOCH_NUMBER, BALANCING_EPOCH_NUMBER
Output: Learned model

```
for epoch ← 1 to EPOCH_NUMBER do
    for batch ← 1 to BATCH_NUMBER do
        Forward propagate
        Reload global features as Equation 11
        for epoch_balancing ← 1 to BALANCING_EPOCH_NUMBER do
            Optimize sample weights as Equation 12
        end for
        Back propagate with weighted prediction loss as Equation 13
        Save features and weights as Equation 14
    end for
end for
```

B. Experiments

B.1. Datasets

We adopt 4 datasets to conduct experiments in our 4 settings. We briefly introduce them as follows.

MNIST-M is generated by the method in the original paper, which is blending digits from the original MNIST dataset over patches extracted from images in BSDS500 dataset.

VLCS consists of 5 object categories shared by the PASCAL VOC 2007, LabelMe, Caltech and Sun datasets. We follow the standard protocol of [18] and divide each domain into a training set (70%) and validation set (30%) randomly.

PACS is a widely used benchmark for domain generalization which consists of 7 object categories spanning 4 image styles, namely *photo*, *art-painting*, *cartoon and sketch*. We adopt the protocol in [36] to split the training and val set.

NICO is dedicately designed for Non-I.I.D (distribution shifts) image classification. The images from each category can be wildly various and labeled with 10 contexts.

B.2. Training Details

Table 5: The structure of CNNs for MNIST-M

Layer	Details
Input	$3 \times 28 \times 28$
Conv	Kernel Size 7, Stride 1, Out Channel 32, BN, ReLU
Conv	Kernel Size 5, Stride 2, Out Channel 32, BN, ReLU
Dropout	$p = 0.4$
Conv	Kernel Size 3, Stride 1, Out Channel 64, BN, ReLU
Conv	Kernel Size 3, Stride 2, Out Channel 64, BN, ReLU
Dropout	$p = 0.4$
FC	Out Channel 16, ReLU
SoftMax	Class_Num

For PACS and VLCS, we adopt ResNet-18[21] as the backbone model. Given the images from PACS and VLCS are not sufficient to train a randomly initialized deep model like ResNet-18, we use the weights pretrained on ImageNet[9]. For NICO, randomly initialized ResNet-18 is used as the backbone model. For PACS and VLCS, We train all the methods 30 epochs. The initial learning rate is 0.01 and decays with a rate of 0.1 at epoch 24. For NICO, we train all the methods 60 epochs. We set initial learning rate to 0.02 which decays with a rate of 0.1 at epoch 30. The weight decay is set to 0.0005 for all the three datasets. For MNIST-M, we use a 4-layer convolutional network as the backbone and the structure is shown in Table 5. The models are trained 30 epochs. The learning rate is 0.02 and decays with a rate of 0.1 at epoch 20. The weight decay is 0.001.

For JiGen [7], DG-MMLD[46] and RSC [27], we adopt the code published by the authors of original papers and the hyperparameters reported in original papers. For M-ADA [52], we train the model for 5000 iterations on MNIST-M, PACS and VLCS, 8000 iterations on NICO.

B.3. More Results and Data Split Details of Unbalanced Setting

In the *unbalanced* setting, , we randomly choose a dominant domain for each target domain for both PACS and VLCS. The ratio of data amount from dominant domain to other domains are 5:1:1 in Section 4.2. To simulate a more general setting, we also evaluate DG methods on settings where domain ratio is 3:1:1 and 2:1:1. The numbers of samples from each domain on PACS are shown in Table 17, 13, and 15. The numbers of samples from each domain on VLCS are shown in Table 18, 14 and 16.

The results of the *unbalanced* setting when the domain ratio is 3:1:1 on PACS and VLCS are shown in Table 6. StableNet outperforms all the other state-of-the-art methods on all the target domains on VLCS and three out of four domains on PACS. StableNet achieves the highest average accuracy across four domains both on PACS and VLCS. The

Table 6: Results of the *unbalanced* setting when the domain ratio is 3:1:1 on PACS and VLCS. The reported results are average over three repetitions of each run. The title of each column indicates the name of the domain used as target. The best results of all methods are highlighted with the bold font.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen [7]	76.25	71.60	68.79	92.05	77.17	84.82	58.45	64.42	56.06	65.94
M-ADA [52]	65.76	66.98	54.49	90.33	69.39	66.25	53.13	43.80	47.30	52.62
DG-MMLD [46]	81.98	73.53	74.32	92.65	80.61	79.42	58.29	64.96	51.16	63.46
Focal loss [41]	75.56	70.13	67.91	90.75	76.09	75.22	53.98	56.61	53.88	59.92
RSC [27]	**84.44**	74.82	70.87	94.66	81.20	82.31	59.55	**65.32**	56.11	65.82
ResNet-18	75.90	73.34	69.22	93.82	78.07	81.66	61.75	60.20	57.33	65.24
StableNet (ours)	79.15	**79.96**	**75.44**	**95.19**	**82.44**	**85.51**	**65.46**	63.65	**59.93**	**68.64**

Table 7: Results of the *unbalanced* setting when the domain ratio is 2:1:1 on PACS and VLCS. For the number of each run and the use of font, see Table 6.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen	74.20	71.60	68.88	93.15	76.96	89.29	64.41	66.38	55.46	68.89
M-ADA	74.32	57.03	59.40	93.67	71.11	87.03	63.52	54.58	52.03	64.29
DG-MMLD	79.94	71.67	68.58	92.85	78.26	86.39	66.67	65.39	54.59	68.26
RSC	80.51	69.53	68.96	**94.36**	78.34	87.59	65.05	65.67	54.37	68.17
ResNet-18	75.75	68.35	64.59	93.37	75.52	84.77	64.62	65.38	52.59	66.84
StableNet (ours)	**80.56**	**75.33**	**71.48**	94.29	**80.42**	**91.29**	**67.87**	**66.72**	**56.55**	**70.61**

results of the *unbalanced* setting when the domain ratio is 2:1:1 on PACS and VLCS are shown in Table 7. StableNet also shows strong ability of generalization consistently in all the three *unbalanced* settings, which indicates the effectiveness of StableNet on more general settings of distribution shift problems.

B.4. More Results and Data Split Details of Unbalanced + Flexible Setting

In the *unbalanced + flexible* setting, we randomly choose a dominant domain for each target domain for both PACS and VLCS. The ratio of data amount from dominant domain to other domains are 5:1:1 in Section 4.3. We evaluate DG methods on *unbalanced + flexible* settings where the domain ratio is 5:1:1, 3:1:1, 2:1:1 and 1:1:1, respectively. Note that when the ratio is set to 1:1:1, this setting degenerates to *flexible* setting where the numbers of samples from different domains are approximately the same.

Actually, the classic DG setting is not sufficient for evaluating the generalization ability of models since the target domain is unique when the source domains are determined. So given the source domains, only the performance on a given target domain is evaluated while the goal of generalization is to generalize to any target domains. The *unbalanced + flexible* setting, however, can evaluate the ability of generalization to any domains since the model are tested on all the domains for given source domains. In other words, the model is trained once and tested on all the domains, while in the classic DG setting, the model is tested on one single domain for each training. Moreover, there are no overlap between source domains and the target domain for a single class so the generalization ability is evaluated.

The results of the *unbalanced + flexible* setting when the domain ratio is 3:1:1 and 2:1:1 on PACS and VLCS are shown in Table 8 and Table 9, respectively. Note that the target domain for each class is randomly chosen so that there are two domains containing no test data. Details of data split of this setting are shown in Table 17 and Table 18. Given the capacities of different target domains can be significantly various, we report the weighted average accuracy (denoted by 'overall' in the table) of all domains instead of naive average accuracy for the *unbalanced + flexible* setting, which is different from all the reported average accuracy for other settings. We show the accuracy of methods on all the domains. StableNet outperforms all the other state-of-the-art methods on almost all the target domains and on average accuracy on PACS and VLCS. Moreover, StableNet achieves the highest average accuracy across four domains both on PACS and VLCS. The results of the *unbalanced + flexible* setting when the domain ratio is 1:1:1 on PACS and VLCS are shown in Table 10, where StableNet also outperforms other state-of-the-art counterparts.

Table 8: Results of the *unbalanced + flexible* setting when the domain ratio is 3:1:1 on PACS and VLCS. For the number of each run and the use of font, see Table 6.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Overall	Caltech	Labelme	Pascal	Sun	Overall
JiGen	57.95	-	27.70	90.02	44.03	97.71	-	61.50	67.12	76.31
M-ADA	49.58	-	15.21	79.92	33.33	97.55	-	44.17	60.32	69.09
DG-MMLD	68.57	-	37.65	91.91	52.56	**99.85**	-	65.37	67.16	78.08
RSC	66.10	-	26.48	90.52	44.55	99.82	-	56.79	65.25	75.09
ResNet-18	57.11	-	28.54	89.79	45.07	99.25	-	63.37	63.55	75.93
StableNet (ours)	**71.82**	-	**38.16**	**92.36**	**54.10**	99.75	-	**67.53**	**68.89**	**79.28**

Table 9: Results of the *unbalanced + flexible* setting when the domain ratio is 2:1:1 on PACS and VLCS. For the number of each run and the use of font, see Table 6.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Overall	Caltech	Labelme	Pascal	Sun	Overall
JiGen	50.14	87.08	16.45	81.55	40.12	7.11	39.55	39.85	31.90	37.99
M-ADA	43.86	81.60	9.88	63.51	32.17	6.02	35.34	29.59	22.22	27.88
DG-MMLD	66.46	91.78	**30.64**	80.07	51.71	10.44	44.57	41.85	**42.86**	41.72
RSC	53.02	88.09	16.59	83.58	41.24	1.20	36.87	26.89	30.00	30.78
ResNet-18	56.62	80.56	20.98	77.58	42.83	45.42	33.72	39.17	23.81	36.49
StableNet (ours)	**67.85**	**92.01**	29.57	**88.92**	**52.68**	**59.88**	**48.15**	**49.58**	**42.86**	**49.27**

B.5. Details about the generation of MNIST-M in the setting of *composional + dominant + flexible + adversarial*

The MNIST-M are generated by blending digit figures from the original MNIST dataset over patches extracted from images in BSDS500 dataset. The backgrounds are cropped from 200 images, resulting in 200 domains. The backgrounds from the same domain may be different given they are randomly cropped from the same image. We generate the adversarial setting by splitting the domains into 10 subsets responding to the classes. We randomly choose 1 subset for 1 class in the training data and choose 1 domain in the subset as the dominant domain. The ratio of the data from dominant domain to the data from other domains varies from 9.5:1 to 1:1. The subset chosen for one class for training is set to another class for testing, as well as the dominant domain.

B.6. NICO

NICO is a dataset designed for distribution shifts problem. There are 19 categories and 10 contexts (domains) for each category. The domains for different category are various. The standard for split of contexts varies for different categories. For instance, some of the context are divided by the background of images such as 'on water' or 'on grass' while some by the posture of objects such as 'running' or 'standing'. Examples of images from NICO are shown in Figure 6.

There is a baseline method called CNBB in the original paper of NICO. We do not report the results of CNBB for the reason that it is designed for AlexNet and we fail to achieve reasonable results in our framework with CNBB. CNBB adopts Tanh function as the activation function and amplifies features from (-1, 1) to approach to {-1, 1} by a quantization loss shown as follows:

$$\mathcal{L}_{\mathrm{II}} = -\sum_{i=1}^{p} \|g_\phi(x_i)\|_2^2 \tag{15}$$

This loss harms ResNet significantly and it is hard to find proper hyperparameters for CNBB with ResNet as the backbone network. Hence, we do not report the results of CNBB.

B.7. Examples of saliency maps

Examples of saliency maps are shown in Figure 7.

The bright lines in saliency maps generated by JiGen demonstrates the effectiveness of the jigsaw puzzle, in which the model focuses more on the margins of any possible puzzles. And the highlight on the object in saliency maps generated by our method show that our model tends to focus on the object instead of the context. Therefore, our method help deep models learn the true connections between features and labels, resulting in models with stronger ability of generalization under distribution shifts.

Table 10: Results of the *unbalanced + flexible* setting when the domain ratio is 1:1:1 on PACS and VLCS. For the number of each run and the use of font, see Table 6.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Overall.	Caltech	Labelme	Pascal	Sun	Overall.
JiGen	61.84	94.37	26.67	87.76	44.91	99.97	17.93	59.84	53.34	69.71
M-ADA	3.06	80.45	22.87	85.38	36.42	89.92	5.75	46.30	41.54	58.02
DG-MMLD	65.77	97.19	**43.44**	89.87	57.07	**100.00**	20.00	60.43	55.12	70.61
RSC	62.64	96.97	23.99	89.75	43.69	**100.00**	15.17	54.03	50.09	66.87
ResNet-18	64.68	95.56	24.39	90.25	44.15	96.25	3.91	49.34	49.30	63.93
StableNet (ours)	**67.66**	**98.52**	41.90	**95.85**	**57.41**	**100.00**	**24.14**	**63.60**	**63.19**	**74.70**

Table 11: Data split details of *unbalanced* setting on PACS dataset when the domain ratio is 5:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Art painting: 2048	Cartoon: 405	Photo: 405	Sketch: 784
Sketch: 3929	Art painting: 779	Cartoon: 779	Photo: 331
Photo: 1670	Art painting: 327	Sketch: 327	Cartoon: 466
Cartoon: 2344	Photo: 463	Sketch: 463	Art painting: 407

Table 12: Data split details of *unbalanced* setting on VLCS dataset when the domain ratio is 5:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Caltech: 991	Labelme: 196	Pascal: 196	Sun: 458
Sun: 2297	Caltech: 350	Labelme: 372	Pascal: 470
Pascal: 2363	Caltech: 448	Sun: 401	Labelme: 370
Labelme:1589	Pascal: 367	Sun: 367	Caltech: 196

Table 13: Data split details of *unbalanced* setting on PACS dataset when the domain ratio is 3:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Art painting: 2048	Cartoon: 678	Photo: 678	Sketch: 784
Sketch: 3929	Art painting: 1217	Cartoon: 1238	Photo: 331
Photo: 1670	Art painting: 551	Sketch: 538	Cartoon: 466
Cartoon: 2344	Photo: 776	Sketch: 761	Art painting: 407

Table 14: Data split details of *unbalanced* setting on VLCS dataset when the domain ratio is 3:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Caltech: 991	Labelme: 327	Pascal: 327	Sun: 458
Sun: 2297	Caltech: 473	Labelme: 583	Pascal: 470
Pascal: 2363	Caltech: 641	Sun: 647	Labelme: 370
Labelme: 1859	Pascal: 616	Sun: 612	Caltech: 196

Table 15: Data split details of *unbalanced* setting on PACS dataset when the domain ratio is 2:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Art painting: 2048	Cartoon: 1020	Photo: 1020	Sketch: 784
Sketch: 3929	Art painting: 1424	Cartoon: 1681	Photo: 331
Photo: 1670	Art painting: 831	Sketch: 715	Cartoon: 466
Cartoon: 2344	Photo: 1136	Sketch: 1062	Art painting: 407

Table 16: Data split details of *unbalanced* setting on VLCS dataset when the domain ratio is 2:1:1. The dominant domain for each target domain is highlighted with the bold font.

Source			Target
Caltech: 991	Labelme: 467	Pascal: 494	Sun: 458
Sun: 2297	Caltech: 627	Labelme: 846	Pascal: 470
Pascal: 2363	Caltech: 855	Sun: 953	Labelme: 370
Labelme: 1859	Pascal: 927	Sun: 914	Caltech: 196

Table 17: Data split details of *unbalanced + flexible* setting on PACS dataset when the domain ratio is 5:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
Dog	**Cartoon: 350**	Art painting: 70	Photo: 70	Sketch: 772
Elephant	**Cartoon: 411**	Art painting: 82	Sketch: 82	Photo: 202
Giraffe	**Photo: 163**	Art painting: 32	Cartoon: 32	Sketch: 753
Guitar	**Photo: 167**	Cartoon: 33	Sketch: 33	Art painting: 184
Horse	**Cartoon: 291**	Photo: 58	Sketch: 58	Art painting: 201
House	**Cartoon: 259**	Art painting: 51	Sketch: 51	Photo: 280
Person	**Art painting: 404**	Cartoon: 80	Photo: 80	Sketch: 160

Table 18: Data split details of *unbalanced + flexible* setting on VLCS dataset when the domain ratio is 5:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
0	**Labelme: 39**	Caltech: 7	Pascal: 7	Sun: 14
1	**Labelme: 592**	Caltech: 86	Sun: 118	Pascal: 489
2	**Pascal: 210**	Caltech: 42	Labelme: 42	Sun: 725
3	**Pascal: 205**	Labelme: 29	Sun: 21	Caltech: 47
4	**Labelme: 606**	Pascal: 121	Sun: 121	Caltech: 609

Table 19: Data split details of *unbalanced + flexible* setting on PACS dataset when the domain ratio is 3:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
Dog	**Cartoon: 389**	Art painting: 129	Photo: 129	Sketch: 772
Elephant	**Cartoon: 457**	Art painting: 152	Sketch: 152	Photo: 202
Giraffe	**Photo: 182**	Art painting: 60	Cartoon: 60	Sketch: 753
Guitar	**Photo: 186**	Cartoon: 61	Sketch: 61	Art painting: 184
Horse	**Cartoon: 324**	Photo: 108	Sketch: 108	Art painting: 201
House	**Cartoon: 288**	Art painting: 95	Sketch: 80	Photo: 280
Person	**Art painting: 449**	Cartoon: 149	Photo: 149	Sketch: 160

Table 20: Data split details of *unbalanced + flexible* setting on VLCS dataset when the domain ratio is 3:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
0	**Labelme: 56**	Caltech: 18	Pascal: 18	Sun: 14
1	**Labelme: 846**	Caltech: 86	Sun: 281	Pascal: 489
2	**Pascal: 300**	Caltech: 83	Labelme: 62	Sun: 725
3	**Pascal: 294**	Labelme: 29	Sun: 21	Caltech: 47
4	**Labelme: 866**	Pascal: 288	Sun: 288	Caltech: 609

Table 21: Data split details of *unbalanced + flexible* setting on PACS dataset when the domain ratio is 2:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
Dog	**Photo: 189**	Art painting: 94	Cartoon: 94	Sketch: 772
Elephant	**Art painting: 255**	Sketch: 127	Cartoon: 127	Photo: 202
Giraffe	**Photo: 182**	Art painting: 90	Cartoon: 90	Sketch: 753
Guitar	**Cartoon: 135**	Art painting: 67	Sketch: 67	Photo: 186
Horse	**Sketch: 816**	Photo: 199	Cartoon: 324	Art painting: 201
House	**Photo: 280**	Art painting: 140	Sketch: 80	Cartoon: 288
Person	**Cartoon: 405**	Sketch: 160	Photo: 202	Art painting: 449

Table 22: Data split details of *unbalanced + flexible* setting on VLCS dataset when the domain ratio is 2:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
0	**Labelme: 56**	Caltech: 27	Sun: 14	Pascal: 231
1	**Caltech: 86**	Labelme: 43	Sun: 43	Pascal: 489
2	**Labelme: 62**	Pascal: 30	Sun: 30	Caltech: 83
3	**Pascal: 294**	Caltech: 47	Labelme: 29	Sun: 21
4	**Pascal: 1049**	Caltech: 524	Sun: 524	Labelme: 866

Table 23: Data split details of *unbalanced + flexible* setting on PACS dataset when the domain ratio is 1:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
Dog	Cartoon: 389	Art painting: 389	Photo: 189	Sketch: 772
Elephant	Cartoon: 457	Art painting: 255	Sketch: 457	Photo: 202
Giraffe	Photo: 182	Art painting: 182	Cartoon: 182	Sketch: 753
Guitar	Photo: 186	Art painting: 184	Sketch: 184	Cartoon: 135
Horse	Cartoon: 324	Photo: 199	Sketch: 324	Art painting: 201
House	Cartoon: 288	Art painting: 288	Sketch: 80	Photo: 280
Person	Art painting: 449	Cartoon: 405	Photo: 432	Sketch: 160

Table 24: Data split details of *unbalanced + flexible* setting on VLCS dataset when the domain ratio is 1:1:1. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
0	Labelme: 56	Caltech: 56	Pascal: 56	Sun: 14
1	Labelme: 846	Caltech: 86	Sun: 652	Pascal: 489
2	Pascal: 100	Caltech: 83	Labelme: 62	Sun: 725
3	Pascal: 94	Caltech: 47	Sun: 21	Labelme: 29
4	Labelme: 866	Pascal: 866	Sun: 866	Caltech: 609

Figure 5: Example Images from MNIST-M

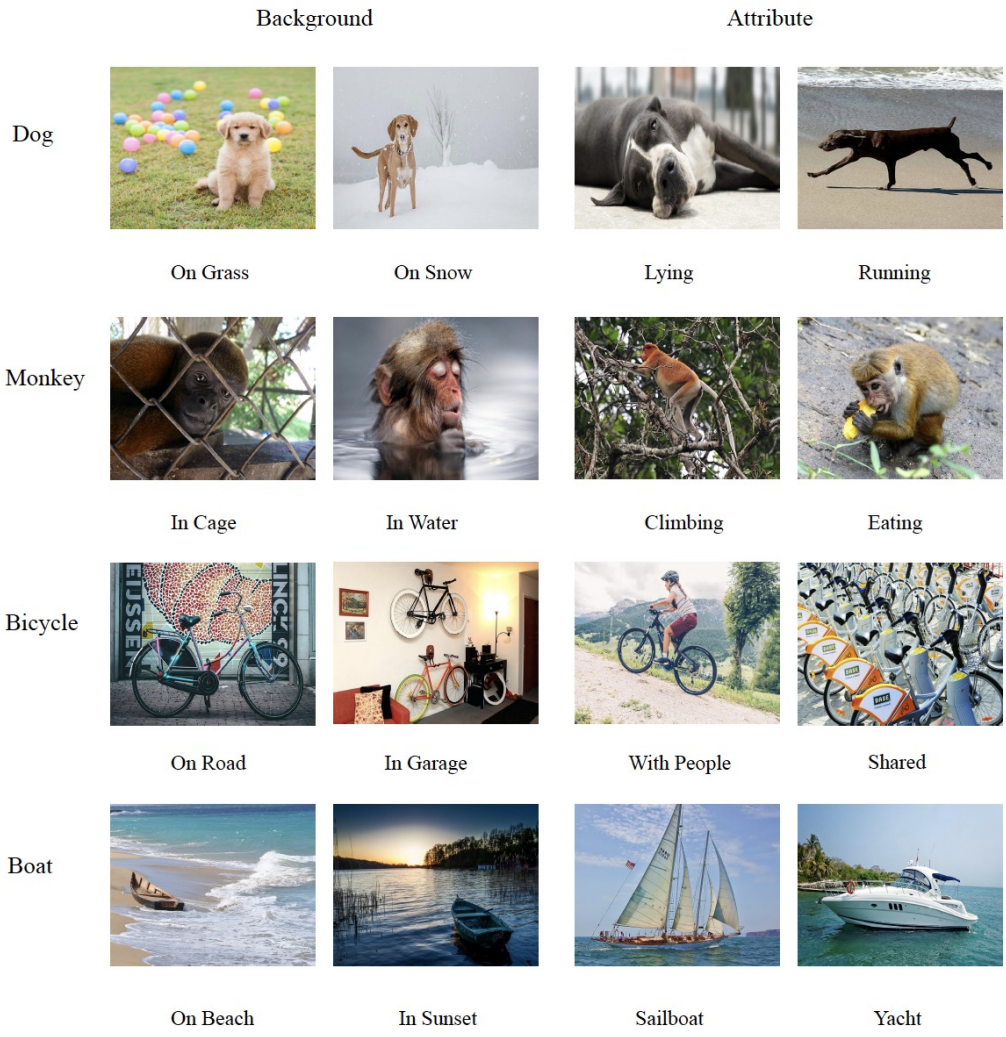


Figure 6: Example images with category and context labels from NICO

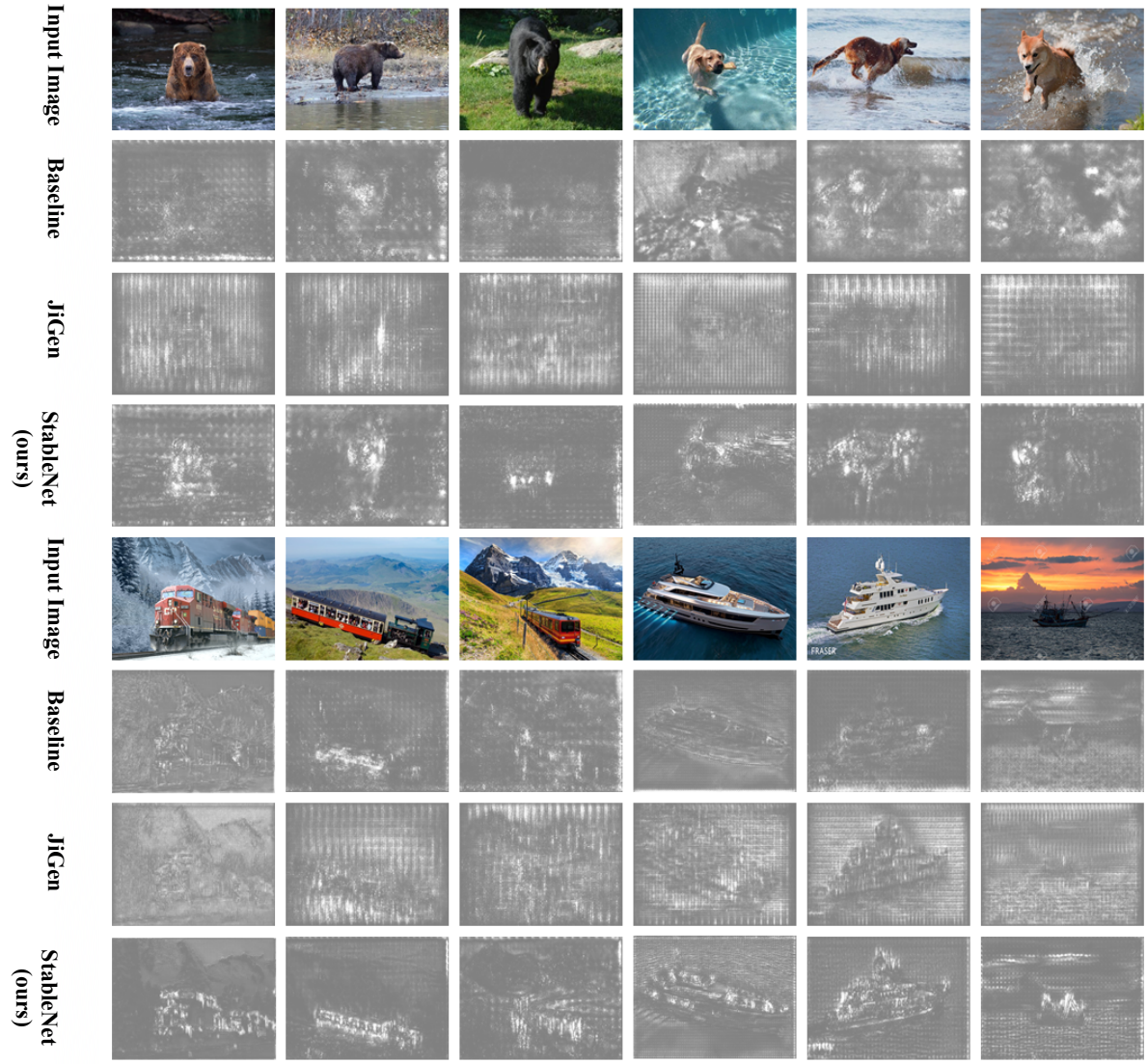


Figure 7: More saliency maps of the ResNet-18 model and StableNet.