SAMPLE REWEIGHTING FOR IMPROVING GENERALIZATION UNDER DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks have achieved striking performance when evaluated on testing data which share the same distribution with training ones, but can significantly fail otherwise. Therefore, eliminating the impact of distribution shifts between training and testing data is of paramount importance for building performance-promising deep models. Conventional methods (e.g. domain adaptation/generalization) assume either the availability of testing data or the known heterogeneity of training data (e.g. domain labels). In this paper, we consider a more challenging case where neither of the above information is available during the training phase. We propose to address this problem by removing the dependencies between features via reweighting training samples, which results in a more balanced distribution and helps deep models get rid of spurious correlations and, in turn, concentrate more on the true connection between features and labels. We conduct extensive experiments on object recognition benchmarks including PACS, VLCS, MNIST-M, and NICO which support the evaluation of generalization ability. The experimental results clearly demonstrate the effectiveness of the proposed method compared with state-of-the-art counterparts.

1 INTRODUCTION

Current machine learning models represent target data encountered during deployment with complex rules learned by minimizing prediction error on training data. Under the assumption that testing and training data share the same distribution, many machine learning approaches have been proposed and shown to be effective. These performance-promising models trained on data with a given distribution tend to exploit subtle statistical correlations among features while refining representations. This exact fitting with training data makes these models more prone to prediction error when there is a distribution shift between training and testing data (Ganin et al. (2016); Zhang et al. (2017)). As a result, selection biases, confounding factors or other peculiarities can make most machine learning models fail to make trustworthy predictions (Arjovsky et al. (2019); Shen et al. (2019)).

A large bunch of existing methods try to address this issue by regularizing the distribution of training data so that it can be more closely aligned with the distribution of testing data (Long et al. (2015); Ganin et al. (2016); Tzeng et al. (2017); Long et al. (2017); Hoffman et al. (2018); Peng et al. (2019)). However, in many real situations, given we have no prior knowledge of testing data which will be generated in the future, these methods are not applicable at such scenarios. Recently, this issue has been intensively studied in *domain generalization (DG)* (Muandet et al. (2013); Ghifary et al. (2015)). The main idea of DG methods is to train the model on samples from several sub-populations labeled with both class label and domain label (Khosla et al. (2012); Li et al. (2018a;b); D'Innocente & Caputo (2018); Li et al. (2019); Jin et al. (2020)). Nevertheless, most datasets are a mixture of multiple latent domains, which can be difficult to know the domain labels (Matsuura & Harada (2020)). Several methods are proposed to address this issue by unsupervisedly generating multiple domains from one single domain (Qiao et al. (2020); Matsuura & Harada (2020)). However, the performance of these methods highly depends on the heterogeneity of the generated domains, which is hardly guaranteed.

In this paper, we address the distribution shifts problem by sample reweighting without knowing either the testing data distribution or the heterogeneity (e.g. domain labels) of training data. The notion underpinning our idea is that a new distribution can be approximated by reweighting the

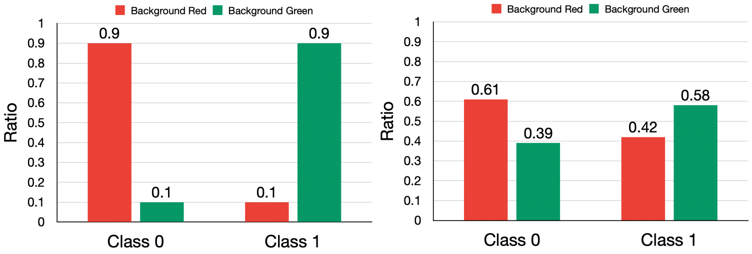


Figure 1: Results of original and balanced distribution. The original distribution is on the left and the balanced one on the right. CNNs can only achieve 25.76% accuracy when trained on original data while our method can balance the distribution and achieve 86.28% accuracy.

samples we observed in training phase. We find the key factor which hinder the generalization of learned model is the unbalanced training distribution, embodied in the complex dependencies and entanglement between features. Such imbalance would make learned model absorb subtle and spurious correlation from training data and fail to generalize onto different distributed testing data.

Take a synthetic case as example, consider the problem of classifying images of hand-written numbers 0 and 1. Instead of adopting conventional gray-scale images, we color the numbers with red and green background. If all the numbers are colored randomly, a deep model can easily achieve the classification accuracy of 99%. However, if we make the factor of color strongly correlated with the label and reverse such correlation between training and testing data, the model fails thoroughly. For instance, in training data, 90% of 0 are with a background of color red, and 10% are with color green, while 90% of number 1 are with green and 10% are with color red. In testing data, only 10% of number 0 are with color red and 90% with color green, 10% of number 1 are with color green and 90% are with color red. Consequently, the spurious correlation between color and label drastically misleads the model and results in the performance even worse than random guesses. Note that the similar scenarios are also common in real world pictures, such as the spurious correlation between grass and cows (Arjovsky et al. (2019); Beery et al. (2018)). In the above example, if we take a look at the training distribution, we can find the color features are highly entangled with truly determining features (e.g. strokes of digits). So a reasonable solution is, if we can adjust the training distribution by a new set of sample weights so that the statistical dependencies between features are removed, such more "balanced" distribution would help predictive models concentrate more on true connections between features and labels.

Recently, there are several methods which also focus on the correlation between features (Kuang et al. (2018); Shen et al. (2020)), but they are all developed under linear models with raw input features and cannot be applied to deep models or more complicated data types (e.g. images and videos). There are two major problems in such more realistic and challenging setting: 1) complex correlations between visual features and category concept brings strong non-linear dependencies among features, the measure and elimination of which are much more complicated than that of a linear one. 2) balancing the distribution of large amounts of data globally in deep models requires excessive storage space and computational cost. On the other hand, trivially adopting a batch balancing scenario would also induce stability issues such as the high variance of sample weights. In this paper, we propose a method called **Sample Reweighted Distribution Balancing (SRDB)**, which balances the training distribution by globally reweighting the training samples and can be integrated into any off-the-shelf deep learning frameworks. In terms of the first problem, we propose to learn any possible correlation including non-linear ones based on Random Fourier Features (Rahimi & Recht (2008)), which limits the computational complexity to an acceptable linear range. As for the second problem, we propose an updating mechanism, called saving and reloading global correlation, to perceive and remove correlation globally by iteratively saving and reloading features and weights of the model in the training phase. The whole method can be jointly optimized and integrated into the existing representation learning pipelines to balance the distribution of training data. After applying SRDB, the models are capable of getting rid of nuisance factors like background colors and therefore can generalize better under distribution shifts, especially the shifts over these nuisance factors. As can be seen in Fig.1, the color distribution is more balanced with the learned sample weights from SRDB and the classification accuracy is boosted by a considerably large margin.

In extensive experiments across a wide range of datasets, our method achieves considerable improvement compared with baseline methods, especially in more challenging settings with strong spurious correlations among training data. The method is also evaluated on real world images with complex contexts and achieves higher accuracy than state-of-the-art baselines.

2 Sample Reweighted Distribution Balancing

We address the distribution shifts problem by reweighting samples globally. Recall the problem of colored hand-written numbers example as we described in Section 1, the key factor of the problem lies in the complex correlation between the nuisance factors like background color and truly determining features like strokes of digits. If the statistical dependencies between nuisance features and determining features are eliminated, the model tends to learn the true relationship between the determining features and the labels, while neglecting the impact of the nuisance features. As a result, the model generalizes better under distribution shifts.

However, distinguishing nuisance features and the determining features requires extra supervisory knowledge. Hence, we propose to directly decorrelate all the features for every input sample to balance the distribution of training data. Our method called Sample Reweighted Distribution Balancing (SRDB) eliminates the dependence between any pair of features for all observed samples. Concretely, SRDB gets rid of all the linear and non-linear dependencies between features by utilizing the characteristics of Random Fourier Features (RFF) and sample reweighting. Moreover, to adapt the global balancing 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 2.1. In Section 2.2, 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. We have n samples $\mathbf{X} \subset \mathbb{R}^{n \times m_X}$ with labels $\mathbf{Y} \subset \mathbb{R}^{n \times m_Y}$. 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 $\mathbf{Z}_{:,i}$.

2.1 Sample Re-weighting with Random Fourier Features

Independence testing statistics To eliminate the dependence between features 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 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 to exploit how relevant the two variables are just from 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} (Fukumizu et al. (2004)) from $\mathcal{H}_B$ to $\mathcal{H}_A$ is defines as follows

$$\langle f, \Sigma_{AB} g \rangle = \mathbb{E}_{AB}\left[f(A)g(B)\right] - \mathbb{E}_A[f(A)]\mathbb{E}_B[g(B)] \tag{1}$$

for all $f \in \mathcal{H}_A$ and $g \in \mathcal{H}_B$. With the operator, the independence can be determined by the following proposition.

Proposition 2.1 (Fukumizu et al. (2008)). *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*

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

Furthermore, Gretton et al. (2008) proposed Hilbert-Schmidt independence criterion, which requires that the squared Hilbert-Schmidt norm of Σ_{AB} should be zero. However, these criteria can hardly be directly applied to measure the level of independence between the random variables. Strobl et al. (2019) noticed that Frobenius norm corresponds to the Hilbert-Schmidt norm in Euclidean space and designed an independent testing statistic, which is introduced as follows.

Let the partial cross-covariance matrix be:

$$\hat{\Sigma}_{AB} = \frac{1}{n-1}\sum_{i=1}^{n}\left[\left(\mathbf{f}(A_i) - \frac{1}{n}\sum_{j=1}^{n}\mathbf{f}(A_j)\right)^T\left(\mathbf{g}(B_i) - \frac{1}{n}\sum_{j=1}^{n}\mathbf{g}(B_j)\right)\right], \tag{3}$$

where

$$\begin{aligned}\mathbf{f}(A) &= (f_1(A), f_2(A), \dots f_{n_A}(A)), \quad f_j(A) \in \mathcal{H}_{\text{RFF}}, \forall j,\\ \mathbf{g}(B) &= (g_1(B), g_2(B), \dots g_{n_B}(B)), \quad g_j(B) \in \mathcal{H}_{\text{RFF}}, \forall j.\end{aligned} \tag{4}$$

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

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

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. Strobl et al. (2019) shows that the independence test will be more accurate as n_A and n_B increase and it is enough to judge the independence of random variables when setting n_A and n_B as 5 in practice.

Sample reweighting for independence Inspired by Kuang et al. (2020), we propose to eliminate the dependence between features in the representation space via sample reweighting 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 reweighting, the partial cross-covariance matrix for random variables A and B in equation 3 should be calculated as follows

$$\hat{\Sigma}_{AB;\mathbf{w}} = \frac{1}{n-1}\sum_{i=1}^{n}\left[\left(w_i\mathbf{f}(A_i) - \frac{1}{n}\sum_{j=1}^{n}w_j\mathbf{f}(A_j)\right)^T\left(w_i\mathbf{g}(B_i) - \frac{1}{n}\sum_{j=1}^{n}w_j\mathbf{g}(B_j)\right)\right]. \tag{6}$$

$\mathbf{f}$ and $\mathbf{g}$ are the RFF mapping functions explained in equation 4. SRDB targets independence between any pair of features. Specifically, for feature $Z_{:,i}$ and $Z_{:,j}$, 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

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

where $\Delta_n = \left\{\mathbf{w} \in \mathbb{R}_+^n \mid \sum_{i=1}^n w_i = n\right\}$. Hence, while using the optimal $\mathbf{w}^*$ to reweight the training samples, the dependence between features can be mitigated to the greatest extent.

2.2 Learning sample weight globally

Equation 7 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:

$$\mathbf{Z}_O = (\mathbf{Z}_{G1}, \mathbf{Z}_{G2}, \cdots, \mathbf{Z}_{Gk}, \mathbf{Z}_L), \qquad \mathbf{w}_O = (\mathbf{w}_{G1}, \mathbf{w}_{G2}, \cdots, \mathbf{w}_{Gk}, \mathbf{w}_L), \tag{8}$$

where $\mathbf{Z}_O$ and $\mathbf{w}_O$ are 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 8 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 7. 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:

$$\mathbf{Z}'_{Gi} = \alpha_i \mathbf{Z}_{Gi} + (1-\alpha_i)\mathbf{Z}_L, \qquad \mathbf{w}'_{Gi} = \alpha_i \mathbf{Z}_{Gi} + (1-\alpha_i)\mathbf{w}_L. \tag{9}$$

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 the sample weight using Equation 7 and the model parameters. While 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 A.

3 Experiments

In this section, we validate SRDB in a variety of settings. The conventional experimental setting of domain generalization (DG) is limited since the distribution shifts caused by the unbalance of capacities of different domains is ignored and domain labels are required. As a result, in the DG setting datasets are split to train, test and val set by domains. There are no overlap in domains between train and test set, and moreover, the available capacities for training of various domains are approximate constant. To cover more general and challenging cases of distribution shifts, we evaluate SRDB in four settings: 1) *compositional*, 2) *compositional + dominant*, 3) *compositional + dominant + flexible*, 4) *compositional + dominant + flexible + adversarial*. The following four subsections explain our objective and details of these settings.

We consider four datasets to carry through these four settings, namely PACS (Li et al. (2017)), VLCS (Torralba & Efros (2011)), MNIST-M (Ganin & Lempitsky (2015)) and NICO (He et al. (2020)). Introduction to these datasets is in appendix B.1.

3.1 Compositional Setting

The *compositional* 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 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 Carlucci et al. (2019); Matsuura & Harada (2020) for both the datasets and utilize three domains as source domains and the remaining one as target.

The results are shown in Table 1. On both PACS and VLCS, SRDB outperforms other state-of-the-art methods, even including those trained with domain labels, in three out of four target cases and achieves the highest average accuracy. As mentioned in Section 1, SRDB can decorrelate visual features and help deep models learn more knowledge of true connection between category labels and visual features related to objects instead of domains.

3.2 Compositional + dominant Setting

In the common DG setting, the capacities of source domains are assumed to be comparable. However, the amount of data that are available from different sources can be various. We simulate this scenario with the *compositional + dominant* setting. To make the amount of data from heterogeneous sources clearly differentiated, we set one domain as the dominant domain. Specifically, same as Section 3.1, three domains are considered as source domains and the other one as target in both PACS and VLCS. For each target domain, we randomly select one domain from the source domains as the dominant source domain and the ratio of data from the dominant domain and the other two domains is 5:1:1. Details of partition are shown in appendix B.2.

The results are shown in Table 2. Compared with the setting in Section 3.1, the performances of most methods drop because the dominant domain interferes with the learning of data from other domains.

Table 1: Results of the *compositional* setting on PACS and VLCS. All the results on PACS are obtained from the original papers of these methods. We implement the methods that require no domain labels on VLCS since these methods are tested with AlexNet (Krizhevsky et al. (2012)) in original papers while we adopt ResNet18 (He et al. (2016)) as the backbone network for all the methods. The results of our method are average over three repetitions of each run. The title of each column indicates the name of the domain used as target. The methods that require domain labels are labelled with asterisk. The best results of all methods are highlighted with the bold font.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
ResNet-18	78.97	77.02	74.51	94.69	81.30	96.77	62.72	72.72	72.68	76.22
JiGen (Carlucci et al. (2019))	79.42	75.25	71.35	96.03	80.51	96.17	62.06	70.93	71.40	75.14
M-ADA (Qiao et al. (2020))	64.29	72.91	67.21	88.23	73.16	74.33	48.38	45.31	33.82	50.46
DG-MMLD (Matsuura & Harada (2020))	81.28	77.16	72.29	96.09	81.83	**97.01**	62.20	73.01	72.49	76.18
D-SAM* (D'Innocente & Caputo (2018))	77.33	72.43	77.83	95.30	80.72	-	-	-	-	-
Epi-FCR* (Li et al. (2019))	82.10	77.00	73.00	93.90	81.50	-	-	-	-	-
FAR* (Jin et al. (2020))	79.30	77.70	74.70	95.30	81.70	-	-	-	-	-
SRDB (ours)	79.44	**78.11**	**78.34**	**96.53**	**83.11**	96.67	**65.36**	**73.59**	**74.97**	**77.65**

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

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
ResNet-18	79.03	68.60	68.45	92.55	77.16	88.61	62.97	61.84	**51.82**	66.31
JiGen	74.53	69.74	66.41	88.12	74.70	87.76	62.52	61.49	48.11	64.97
M-ADA	61.83	69.48	59.46	83.28	68.51	71.09	54.83	47.92	36.72	52.64
DG-MMLD	77.97	70.67	65.39	92.95	76.75	83.16	65.14	62.84	44.40	63.89
SRDB (ours)	**84.52**	**74.68**	**71.30**	**94.26**	**81.19**	**88.73**	**66.49**	**64.68**	51.75	**67.91**

As the intervention grows, our method achieves increasingly improvements compared with baseline methods. As demonstrated in Table 2, our method outperforms other method in all the target domains on PACS and three out of four domains on VLCS.

3.3 COMPOSITIONAL + DOMINANT + FLEXIBLE SETTING

In this subsection, we consider a more challenging setting where domains for different categories can be various. For instance, dogs can be on grass but hardly in water while fishes are otherwise. If we consider the backgrounds in images as an indicator of domain division, images for class 'dog' can be divided into domain 'on grass' but cannot into domain 'in water' while images for class 'fish' are otherwise, resulting in the diversity of domains among different classes. So 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. Hence, we evaluate generalization methods in this subsection when the distribution shifts are not constant for various classes.

We adopt PACS, VLCS and NICO for evaluation. For PACS and VLCS, we randomly select one domain as the dominant domain for each class, and another domain as the target. The dominant ratio is the same as that in Section 3.2. For NICO, there are 10 domains for each class, 8 out of which are selected as the source and 2 as the target. There is a dominant domain and the ratio of the dominant domain to others is 3:1. Details about the data division are shown in appendix B.2.

The results are shown in Table 3. Conventional methods other than JiGen fail to outperform ResNet-18 on NICO under this setting. M-ADA, which generates images for training with an autoencoder, may break down when the input is 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. In contrast, SRDB shows a strong ability of generalization when the input data are with complicated structure especially real-world images from unlimited resources. SRDB can capture various forms of dependencies and balance the distribution of input data. On PACS and VLCS, SRDB 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 constant.

Table 3: Results of the *compositional + dominant + flexible* setting on PACS, VLCS and NICO.

	ResNet-18	JiGen	M-ADA	DG-MMLD	SRDB (ours)
PACS	48.43	51.38	34.90	54.07	**59.84**
VLCS	75.31	72.19	70.31	76.65	**78.88**
NICO	51.71	54.42	40.78	47.18	**59.76**

Table 4: Results of the *compositional + dominant + 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	DR0.95	DR0.9	DR0.8	DR0.7	DR0.6	DR0.5
CNNs	96.93	93.76	91.93	88.13	81.48	68.43	66.11
JiGen	97.18	94.97	92.99	90.64	78.97	68.79	69.34
M-ADA	95.92	94.45	92.29	88.87	85.89	70.32	67.08
DG-MMLD	96.89	94.61	92.59	89.72	**88.44**	69.13	71.39
SRDB (ours)	**97.35**	**95.33**	93.49	91.24	87.04	**75.69**	**75.46**
SRDB (ours) + JiGen	97.29	94.96	**94.63**	**91.62**	85.94	72.83	71.11

3.4 Compositional + dominant + flexible + adversarial Setting

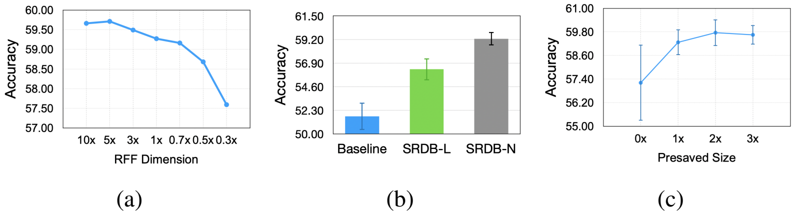
In the most challenging scenario, the spurious correlations between domains and labels are strong and misleading. For instance, we assume a scenario where the category 'dog' is usually associated with the domain 'grass' and the category 'cat' with the domain 'sofa' in the training data, while the category '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. To exploit the effect of various levels of domain shifts, 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 dominant domain 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. More information about the method of data generating and sample images are in Appendix B.

The esults are shown in Table 4. As the dominant ratio increases, the spurious correlation between domains and categories becomes stronger so that the performance of predictive models drops. Also, when the imbalance in visual features is significant, our method achieves noticeable improvement compared with baseline methods. Besides, as a simple sample reweighting method that requires no extra supervision, SRDB can be easily integrated into deep predictive frameworks, such as JiGen. We show the effectiveness of the integrated model in the last row of Table 4.

3.5 Ablation study

SRDB 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 Tanskanen et al. (2018), one can further reduce the feature dimension by randomly selecting features used to calculate dependence with different ratios. Figure 2 shows the results of GNLB with different dimensions of random Fourier features.

If we remove all the random Fourier features, our regularizer in Equation 7 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.



(a) (b) (c)

Figure 2: 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%. *SRDB-N* and *SRDB-L* indicate the original SRDB and the degenerated version of SRDB 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.

We further exploit the effect of the size of presaved features and weights in Equation 8 and the results are shown in Figure 2(c). Generally, the accuracy raises slightly as the presaving size increases.

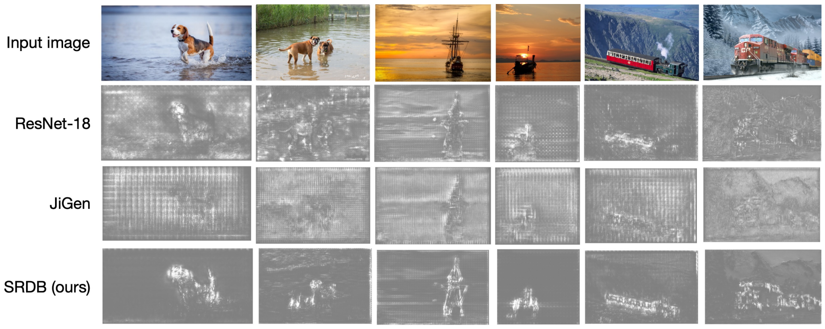


Figure 3: Saliency maps of the ResNet-18 model and the model trained with SRDB. The brighter the pixel is, the more contributions it makes to prediction.

3.6 Saliency Map

An intuitive type of explanation for image classification models is to identify pixels that have a strong influence on the final decision (Smilkov et al. (2017)). 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 Adebayo et al. (2018), which generates saliency maps depending on the learned parameters of the models instead of the architecture.

Visualization results are shown in Figure 3. 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, which further demonstrates that the model trained with SRDB focus more on visual parts which are both distinguishing and invariant when the postures or positions of objects vary.

4 Conclusion

In the paper, to improve the generalization of deep models under distribution shifts, we proposed a novel method called sample reweighed distribution balancing (SRDB) which can balance the distribution of training data via sample reweighting. Extensive experiments across a wide range of settings demonstrated the effectiveness of our method.

REFERENCES

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*, pp. 9505–9515, 2018.

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

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.

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*, pp. 2229–2238, 2019.

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

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.

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

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

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.

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*, pp. 2551–2559, 2015.

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*, pp. 585–592, 2008.

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.

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

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. PMLR, 2018.

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

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

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

Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1617–1626, 2018.

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

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*, pp. 5542–5550, 2017.

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*, pp. 1446–1455, 2019.

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, 2018a.

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, 2018b.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017.

Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, pp. 11749–11756, 2020.

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

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*, pp. 12556–12565, 2020.

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

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. *arXiv preprint arXiv:1911.12580*, 2019.

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. In *AAAI*, pp. 5692–5699, 2020.

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

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.

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.

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

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2020–2030, 2017.

A Detailed Procedure of SRDB

In the training phase, SRDB 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. For a batch of input data, the visual features are extracted by the convolutional layers of the deep model. Then the sample weights are optimized by Equation 7. The conduct of weights and penalties for samples is the final loss used to optimize the convolutional layers as well as the classifier. The present features and weights are integrated with the previous global features and weights as Equation 8 indicates.

In the inference phase, given the backpropagation is disabled, SRDB escapes the reweighting phase and conduct prediction directly.

Algorithm 1 *Sample Reweighted Distribution Balancing (SRDB)*

```
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 and weights
        for epoch_balancing ← 1 to BALANCING_EPOCH_NUMBER do
            Optimize sample weights as Equation 7
        end for
        Back propagate with weighted prediction loss
        Save features and weights as Equation 9
    end for
end for
```

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 in most experiments.

B Detailed experimental settings

In some of our settings, we generate distribution shift between training and testing data by blending objects over different backgrounds, which depends on various of colors. Hence, the data augmentations related to colors may offset the distribution shift. Moreover, we find in experiments that some of the data augmentation approaches contributes differently to different stable learning methods. So we train all the models without any data augmentation on MNIST-M and data augmentations are totally the same for all the methods on other datasets.

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 (Ghifary et al. (2015)) 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 (Li et al. (2017)) 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 Details of data split

In the setting of *compositional + dominant* on PACS and VLCS, we randomly choose a dominant domain for each target domain. The ratio of data amount from dominant domain to other domains are 5:1:1. The numbers of each domain on PACS are shown in Table 5. The numbers of each domain on VLCS are shown in Table 6.

Table 5: Data split details of *compositional + dominant* setting on PACS dataset. 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 6: Data split details of *compositional + dominant* setting on VLCS dataset. 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 7: Data split details of *compositional + dominant + flexible* setting on PACS dataset. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
Dog	**Cartoon: 389**	Art painting: 77	Photo: 77	Sketch: 772
Elephant	**Cartoon: 457**	Art painting: 91	Sketch: 91	Photo: 202
Giraffe	**Photo: 182**	Art painting: 35	Cartoon: 35	Sketch: 753
Guitar	**Photo: 186**	Cartoon: 36	Sketch: 36	Art painting: 184
Horse	**Cartoon: 324**	Photo: 64	Sketch: 64	Art painting: 201
House	**Cartoon: 288**	Art painting: 56	Sketch: 56	Photo: 280
Person	**Art painting: 449**	Cartoon: 89	Photo: 89	Sketch: 160

B.3 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 4.

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:

$$\mathcal{L} = -\sum_{i=1}^{p} \|g_{\phi}(x_i)\|_2^2 \tag{10}$$

Table 8: Data split details of *compositional + dominant + flexible* setting on VLCS dataset. The dominant domain for each target domain is highlighted with the bold font.

Class	Source			Target
0	**Labelme: 56**	Caltech: 10	Pascal: 10	Sun: 14
1	**Labelme: 846**	Caltech: 86	Sun: 86	Pascal: 489
2	**Pascal: 300**	Caltech: 60	Labelme: 60	Sun: 725
3	**Pascal: 294**	Labelme: 29	Sun: 29	Caltech: 47
4	**Labelme: 866**	Pascal: 173	Sun: 173	Caltech: 609

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.

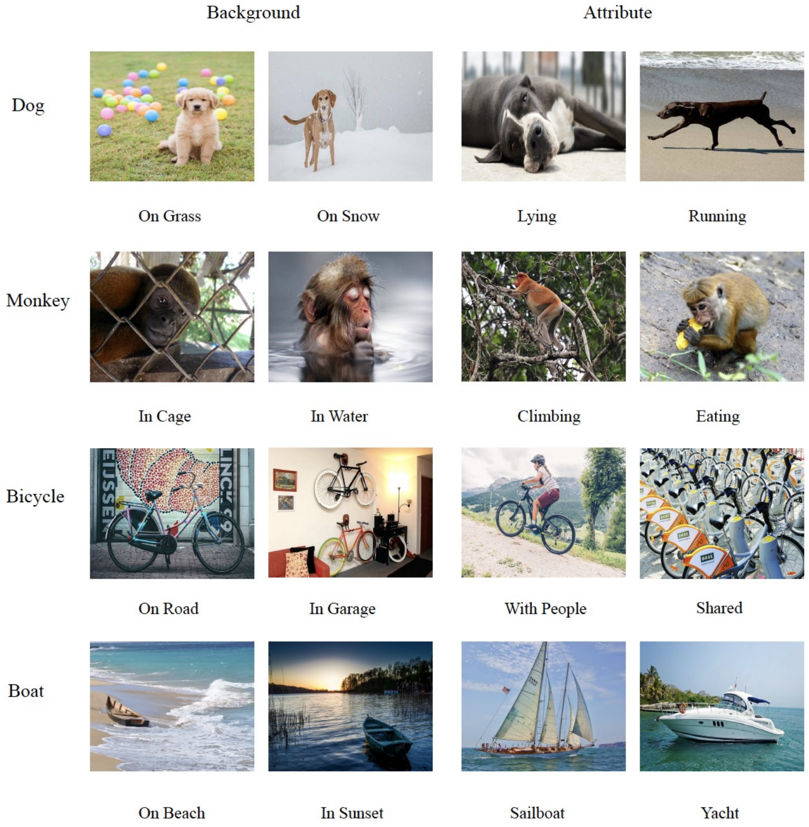


Figure 4: NICO

B.4 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.

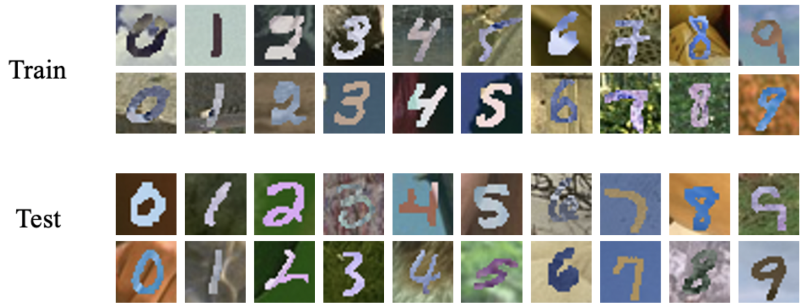


Figure 5: Example Images for MNIST-M

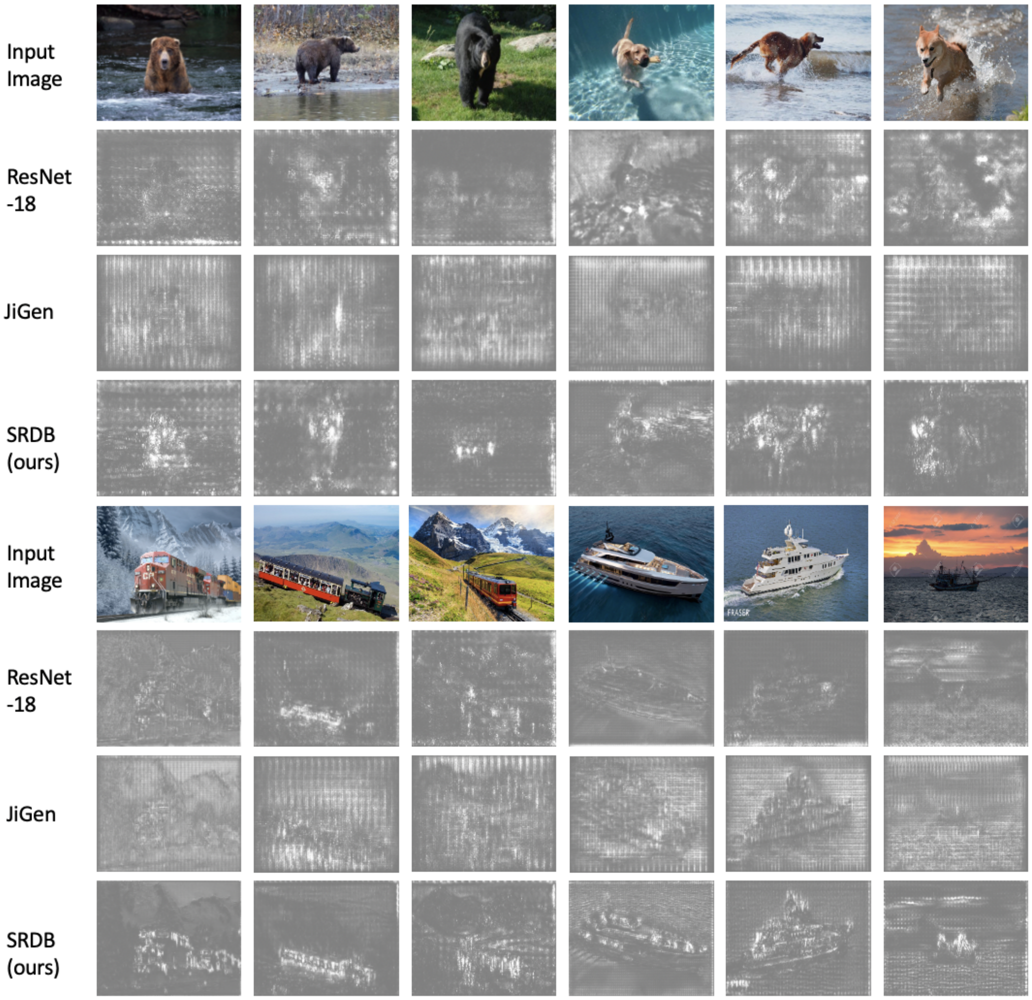


Figure 6: More saliency maps of the ResNet-18 model and the model trained with SRDB.

C Examples of saliency maps

Examples of saliency maps are shown in Figure 6.

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.