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.

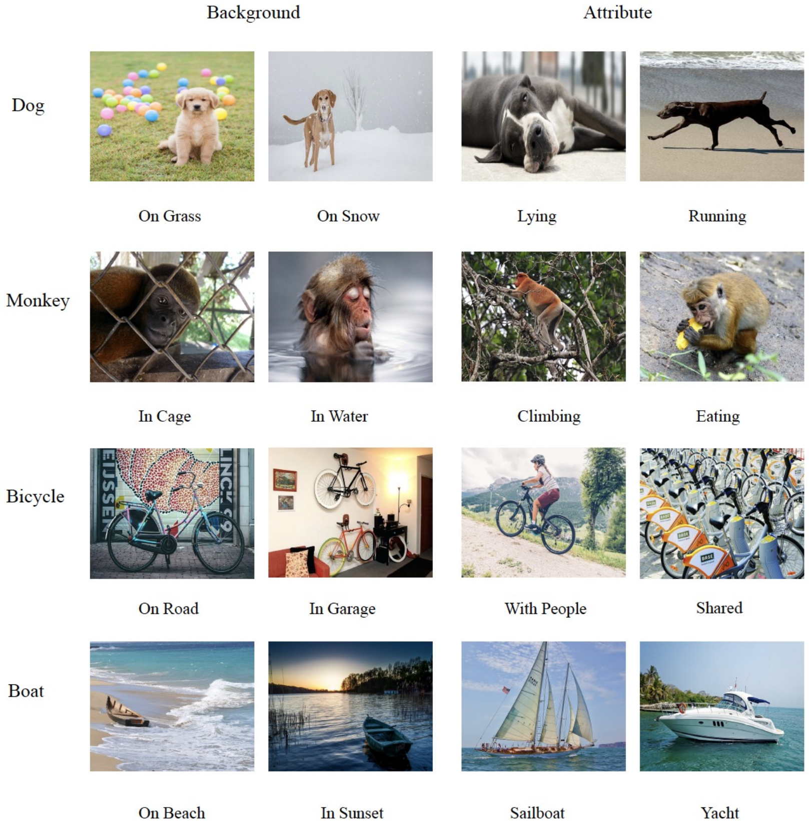

Figure 4: NICO

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.

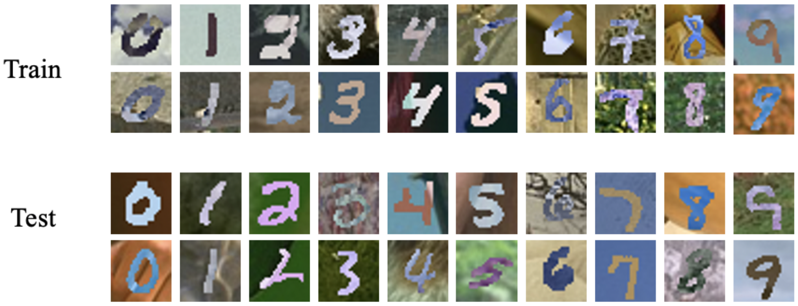

Figure 5: Example Images for MNIST-M

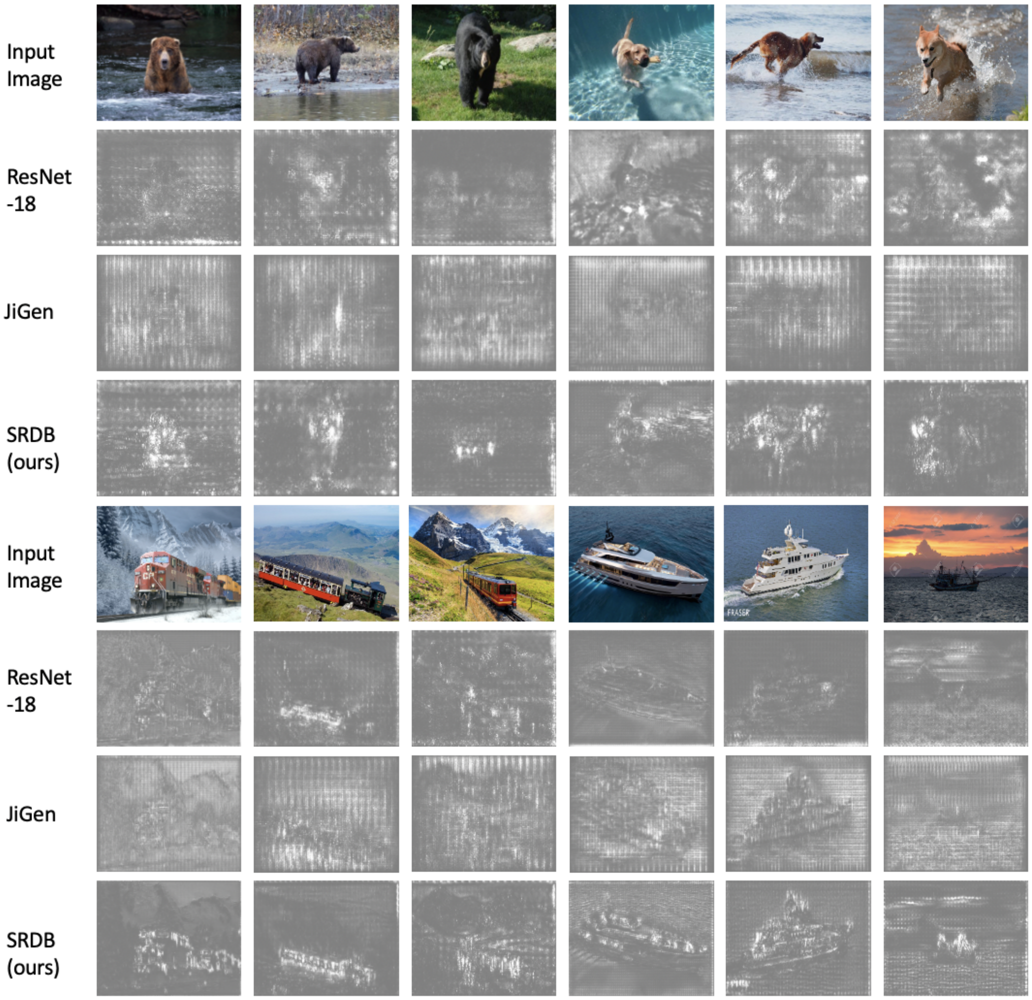

Figure 6: More saliency maps of the ResNet-18 model and the model trained with SRDB.

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.