Table 11. Comparison between InfoDrop and other state-of-the-art results of multi-source domain generalization on PACS dataset. We use * to denote the setting in Carlucci et al. (2019) (*e.g.* extra data augmentation, different train-test split, and different learning rate scheduling). Reported state-of-the-art methods include DSN (Bousmalis et al., 2016), LCNN (Li et al., 2017), MLDG (Li et al., 2018a), Fusion (Mancini et al., 2018), MetaReg (Balaji et al., 2018), JiGen (Carlucci et al., 2019), HEX (Wang et al., 2019b) and PAR (Wang et al., 2019a). We use PAR_B, PAR_M, PAR_H to denote PAR with broader local pattern, more powerful pattern classifier and higher level of local concept, respectively. For more details, please refer to the original paper (Wang et al., 2019a).

	Domain ID	Data Aug.	Art	Cartoon	Photo	Sketch	Average
AlexNet	✗	✗	63.3	63.1	87.7	54	67.03
DSN	✓	✗	61.1	66.5	83.2	58.5	67.33
L-CNN	✓	✗	62.8	66.9	89.5	57.5	69.18
MLDG	✓	✗	63.6	63.4	87.8	54.9	67.43
Fusion	✓	✗	64.1	66.8	90.2	**60.1**	70.30
MetaReg	✓	✗	**69.8**	**70.4**	**91.1**	59.2	**72.63**
HEX	✗	✗	66.8	69.7	87.9	56.3	70.18
PAR	✗	✗	**66.9**	67.1	88.6	62.6	71.30
PAR_B	✗	✗	66.3	67.8	87.2	61.8	70.78
PAR_M	✗	✗	65.7	68.1	88.9	61.7	71.10
PAR_H	✗	✗	66.3	**68.3**	**89.6**	**64.1**	**72.08**
JiGen*	✗	✓	67.6	**71.7**	89.0	65.1	73.38
PAR*	✗	✓	68.0	71.6	**90.8**	61.8	73.05
$\text{PAR}_\text{B}^\star$	✗	✓	67.6	70.7	90.1	62.0	72.59
$\text{PAR}_\text{M}^\star$	✗	✓	68.7	71.5	90.5	62.6	73.33
$\text{PAR}_\text{H}^\star$	✗	✓	68.7	70.5	90.4	64.6	73.54
InfoDrop*	✗	✓	**70.3**	**71.7**	90.3	**70.6**	***75.73***

Table 12. Robustness against 18 types of image corruptions. With vanilla CNN or adversarially trained CNN as baseline, InfoDrop can improve robustness against most corruptions consistently.

Corruption type	Vanilla	+ InfoDrop	+ Adv. Train	+ Info & Adv
Gaussian noise	66.38	69.58	75.30	**76.17**
Shot noise	62.85	66.83	73.80	**74.90**
Impulse noise	49.97	53.00	70.71	**72.26**
Defocus blur	**65.97**	62.52	61.53	62.32
Motion Blur	**74.79**	71.76	71.68	71.32
Zoom blur	**62.92**	58.56	61.58	60.29
Snow	53.10	56.44	61.11	**61.69**
Frost	67.09	**69.80**	69.06	**69.83**
Fog	72.42	**72.75**	54.52	55.00
Brightness	82.20	**82.72**	79.08	78.33
Contrast	**76.66**	75.07	57.93	57.96
Elastic Transform	**76.58**	74.54	71.69	70.26
Pixelate	79.53	**79.81**	78.51	77.66
Jpeg compression	79.77	**80.49**	79.31	78.10
Speckle noise	66.19	69.54	74.74	**75.66**
Gaussian blur	**78.75**	77.03	73.77	74.04
Spatter	79.18	**79.66**	78.04	75.55
Saturate	77.15	**77.77**	72.62	71.26

Table 13. InfoDrop's improvement in absolute accuracies on single-source domain generalization.

SOURCE \ TARGET	PHOTO	ART	CARTOON	SKETCH
PHOTO	*99.88* → *99.82*	**66.21 → 68.70**	**24.15 → 30.67**	**33.60 → 48.36**
ART	**96.71 → 96.83**	*96.46* → *96.66*	**59.77 → 61.22**	**56.35 → 57.16**
CARTOON	86.41 → 85.57	69.29 → 68.85	*99.53* → *99.57*	**64.85 → 69.66**
SKETCH	**32.34 → 44.25**	**27.34 → 31.57**	**43.81 → 50.00**	*99.47* → *99.62*

A. Additional Results

A.1. Results on Domain Generalization and Comparison with Other Shape-biased Models

Several shape-biased methods have recently been proposed to learn robust representations under different domains (Carlucci et al., 2019; Wang et al., 2019a;b). For a full comparison with these state-of-the-art methods, we also test performance of InfoDrop on domain generalization with AlexNet (Krizhevsky et al., 2012) as backbone. We follow the setting in Wang et al. (2019a). Results are shown in Table 11. On all four domains, InfoDrop with vanilla AlexNet as baseline is already better than or comparable to other state-of-the-art methods. Note that among these methods, JiGen, HEX and PAR are methods which explicitly train a shape-biased model.

The absolute accuracies of single-source domain generalization are also postponed here (Table 13) due to the limited space in Table 2.

A.2. Robustness Against Image Corruption

Complete results of robustness against image corruption are shown in Table 12. As baseline, we use both vanilla CNN and adversarially trained CNN. Then we apply InfoDrop and report the improved results. As shown in the table, on both baselines InfoDrop can improve robustness against most corruptions non-trivially.

A.3. Visualization of Saliency Map

Additional visualization results of saliency map of InfoDrop are plotted in Fig. 6. For comparison, saliency map of vanilla CNN is also displayed. Obviously, saliency of InfoDrop is more biased towards global structure, thus more human-aligned and interpretable.

A.4. Visualization of Self-Information

We further visualize self-information on the dataset of Stylized-Imagenet (Geirhos et al., 2019). As a comparison, we also show the results of edge detecting. As shown in Fig. 7 (Top), the original image is stylized with different art work. As a result (Middle), edge detecting is largely influenced by texture information in different style, sometimes even ruining the image content severely. However, distribution of self-information in each stylized image keeps mostly the same, accentuating global structure and meanwhile repressing local texture.

B. Experimental Settings

Of all hyper-parameters, we find r_0 and T the most important for model performance. For all the tasks, we search r_0 in $[0.1, 2.0]$ and T in $[0.01, 1.0]$. We fix $h = 1$, $R = 3$ through the whole experiment. For ResNet18 (He et al., 2016), we apply InfoDrop in both the first convolutional layer and the first residual block, or just in the first layer under some settings. All hyper-parameters are selected according to results on validation set. We use PyTorch (Paszke et al., 2019) for implementation and train all the models on single NVIDIA Tesla P100 GPU.

B.1. Domain Generalization

B.1.1. Dataset

We use PACS (Li et al., 2017) as our dataset for domain generalization. PACS consists of four domains (photo, art painting, cartoon and sketch), each containing 7 categories (dog, elephant, giraffe, guitar, horse, house and person). The dataset is created by intersecting classes in Caltech-256 (Griffin et al., 2007), Sketchy (Sangkloy et al., 2016), TU-Berlin (Eitz et al., 2012) and Google Images. Dataset can be downloaded from `http://sketchx.eecs.qmul.ac.uk/`. Following protocol in Li et al. (2017), we split the images from training domains to 9 (train) : 1 (val) and test on the whole target domain. We use a simple data augmentation protocol by randomly cropping the images to 80-100% of original sizes and randomly apply horizontal flipping.

B.1.2. Parameter Setup

We use ResNet18 (He et al., 2016) as our backbone. Models are trained with SGD solver, 100 epochs, batch size 128. Learning rate is set to 0.001 and shrinked down to 0.0001 after 80 epochs. Bandwidth h and radius R are fixed at 1 and 3, respectively. For photo as source domain, we set $r_0 = 1.5$ and $T = 0.03$. For art or cartoon as source domain, we set $r_0 = 1.5$ and $T = 0.01$. For sketch as source domain, we

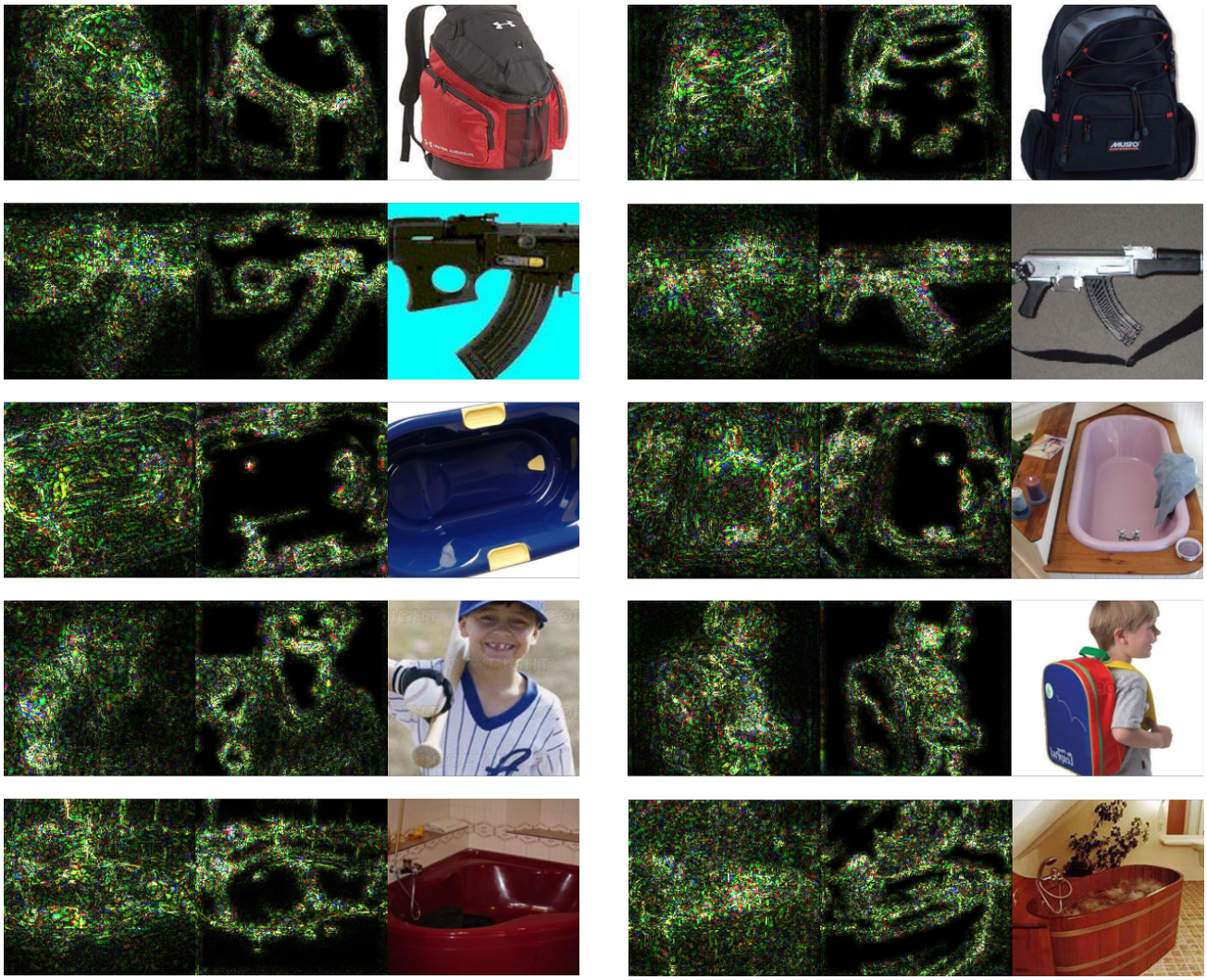

Figure 6. Visualization of CNN's saliency (gradient) map. For each subfigure, from left to right: saliency map of vanilla CNN, saliency map of CNN with InfoDrop and original image.

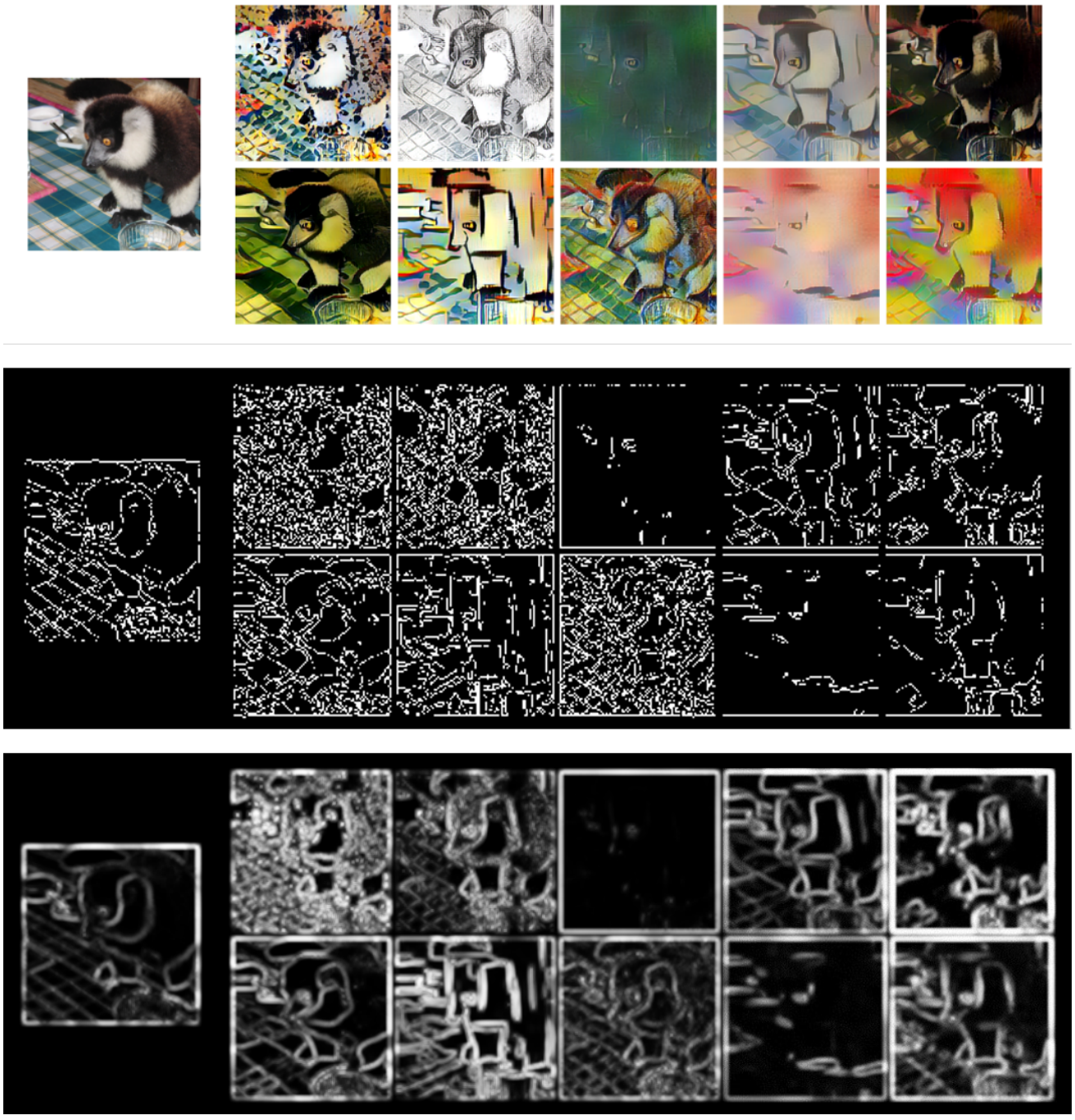

Figure 7. Top: The original image and 10 different stylized version. Stylized images in the example can be found in https://github.com/rgeirhos/Stylized-ImageNet. *Middle*: Edge detecting results of stylized images. *Bottom*: Distribution of self-information in each image.

set $r_0 = 1.2$ and $T = 1.0$.

B.2. Few-shot Classification

B.2.1. Dataset

We mainly use *mini*-Imagenet (Ravi & Larochelle, 2017) and CUB (Wah et al., 2011) as dataset for few-shot classification. Downloadable links of both dataset can be found in this repository https://github.com/wyharveychen/CloserLookFewShot.

mini-Imagenet contains a subset of 100 classes from the whole ImageNet dataset (Deng et al., 2009) and contains 600 images for each class.Following settings in previous works (Ravi & Larochelle, 2017), we randomly divide the whole 100 classes into 64 training classes,16 validation classes and 20 novel classes.

CUB (abbreviation for CUB-200-2011) dataset contains 200 classes with 11788 images. We divide it into 100 base classes, 50 validation classes and 50 novel classes following Hilliard et al. (2018).

We also test our models on the cross-domain scenario, namely *mini*-Imagenet→CUB, where *mini*-ImageNet is used as our base class and the 50 validation and 50 novel classes come from CUB.

Following Chen et al. (2019), we apply data augmentation including random crop, horizontal flip and color jitter.

B.2.2. Parameter Setup

We use 4-layer convolutional neural network (Conv-4) as our backbone, following (Snell et al., 2017). All methods are trained from scratch and use the Adam optimizer with initial learning rate 10^{-3}. In meta-training stage, we train 60000 episodes for 5-way 5-shot classification without data augmentation, and 80000 episodes for 5-way 1-shot classification without data augmentation. When data augmentation is applied, we add an extra 20000 episodes in meta-training stage. In each episode, we sample 5 classes to form 5-way classification. For each class, we pick k labeled instances as our support set and 16 instances for the query set for a k-shot task. Drop coefficient r_0, temperature T, bandwidth h and radius R are fixed at 0.1, 0.5, 1 and 3, respectively. InfoDrop is applied in first two convolutional layers for Conv-4 network, which we use as the backbone through all experiments.

In the fine-tuning or meta-testing stage for all methods, we average the results over 600 experiments. In each experiment, we randomly sample 5 classes from novel classes, and in each class, we also pick k instances for the support set and 16 for the query set. For other settings, we follow the protocol in Chen et al. (2019).

Finally, it is worth noting that since we use the re-implementation in Chen et al. (2019), results of baseline methods may be higher than reported in their original papers. Please refer to Chen et al. (2019) for more details.

B.3. Robustness against Image Corruption

B.3.1. Dataset

For clean images, we use Caltech-256 (Griffin et al., 2007) as dataset. It consists of 257 object categories containing a total of 30,607 images with high resolution. Dataset can be downloaded from http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html. We manually split 20% of images as the test set. Rescaling and random cropping are used as data augmentation following the protocol in He et al. (2016).

For generation of corrupted images, we use the library provided in Hendrycks & Dietterich (2019). Original code for corruption generation can be found in https://github.com/hendrycks/robustness/tree/master/ImageNet-C/imagenet_c. The repository contains 18 types of corruptions: 'gaussian noise', 'shot noise', 'impulse noise', 'defocus blur', 'motion blur', 'zoom blur', 'snow', 'frost', 'fog', 'brightness', 'contrast', 'elastic transform', 'pixelate', 'jpeg compression', 'speckle noise', 'gaussian blur', 'spatter', 'saturate'. The repository provides 5 different levels of corruption severity. In our experiments, we use the highest level, *i.e.*, level-5 severity.

B.3.2. Parameter Setup

We train all models for 10 epochs. We use SGD with learning rate 0.01 for 5 epochs, 0.001 for 3 epochs and 0.0001 for 2 epochs. Through all experiments, we only apply InfoDrop to the first convolutional layer before all residual blocks of ResNet18. Bandwidth h and radius R are fixed at 1 and 3, respectively. For InfoDrop applied on baseline model, we set $r_0 = 0.7$ and $T = 0.3$. For InfoDrop applied together with adversarial training, we set $r_0 = 1.5$ and $T = 0.03$. For adversarial training, we use 20 runs of PGD attack (Madry et al., 2018) with l_∞ norm of $1/255$. Here we use a relatively small norm to simulate the situation where severity of corruption may exceed the norm of adversarial training. Note that we mainly evaluate InfoDrop's incremental effect on baseline and adversarial methods, while not directly comparing InfoDrop with adversarial training.

B.4. Adversarial Robustness

B.4.1. Dataset

For evaluation of adversarial robustness, we use two datasets separately, *viz.* Caltech-256 and CIFAR10. For Caltech-256,

as in B.3.1, we manually split 20% of images as the test set and use rescaling and random cropping for data augmentation. For CIFAR10, we adopt the protocol in Zhang et al. (2019).

B.4.2. Parameter Setup

For experiments on Caltech-256, we train all models for 10 epochs. We use SGD with learning rate 0.01 for 5 epochs, 0.001 for 3 epochs and 0.0001 for 2 epochs. We apply InfoDrop to the first convolutional layer and first residual block of ResNet18. Bandwidth h and radius R are fixed at 1 and 3, respectively. For InfoDrop applied on baseline model, we set $r_0 = 1.5$ and $T = 0.3$. For InfoDrop applied together with adversarial training, we set $r_0 = 2.0$ and $T = 0.01$. For adversarial training, we use 20 runs of PGD attack (Madry et al., 2018).

For experiments on CIFAR10, we follow the protocol in Zhang et al. (2019). We train models for 105 epochs as a common practice. The learning rate is set to $5e-2$ initially, and is reduced by 10 times at epoch 79, 90 and 100, respectively. We use a batch size of 256, a weight decay of $5e-4$ and a momentum of 0.9 for both algorithm. For adversarial attacks, we use 20 runs of PGD with l_∞ norm of $8/255$ and step size of $2/255$. We apply InfoDrop only on the first convolutional layer of ResNet18. We set $r_0 = 1.2$, $T = 0.01$, $h = 1$, $R = 3$.

B.5. Shape-bias of InfoDrop

In the plotting of CNN's saliency map and experiments of patch shuffling, we use photo-domain in PACS as our dataset and adopt the same settings as in domain generalization. In style transfer, we use pretrained ResNet18 and finetune on content and style images from the repository `https://github.com/xunhuang1995/AdaIN-style`.