

# Reducing Domain Gap by Reducing Style Bias

Hyeonseob Nam\* HyunJae Lee\* Jongchan Park Wonjun Yoon Donggeun Yoo  
Lunit Inc.

{hsnam, hjlee, jcpark, wonjun, dgyoo}@lunit.io

## Abstract

*Convolutional Neural Networks (CNNs) often fail to maintain their performance when they confront new test domains, which is known as the problem of domain shift. Recent studies suggest that one of the main causes of this problem is CNNs' strong inductive bias towards image styles (i.e. textures) which are sensitive to domain changes, rather than contents (i.e. shapes). Inspired by this, we propose to reduce the intrinsic style bias of CNNs to close the gap between domains. Our Style-Agnostic Networks (SagNets) disentangle style encodings from class categories to prevent style biased predictions and focus more on the contents. Extensive experiments show that our method effectively reduces the style bias and makes the model more robust under domain shift. It achieves remarkable performance improvements in a wide range of cross-domain tasks including domain generalization, unsupervised domain adaptation, and semi-supervised domain adaptation on multiple datasets.<sup>1</sup>*

## 1. Introduction

Despite the huge success of Convolutional Neural Networks (CNNs) fueled by large-scale training data, their performance often degrades significantly when they encounter test data from unseen environments. This phenomenon, known as the problem of domain shift [41], comes from the representation gap between training and testing domains. For a more reliable deployment of CNNs to ever-changing real-world scenarios, the community has long sought to make CNNs robust to domain shift under various problem settings such as Domain Generalization (DG) [29, 9, 4, 8], Unsupervised Domain Adaptation (UDA) [33, 11, 46, 34, 42], and Semi-Supervised Domain Adaptation (SSDA) [7, 50, 44].

In stark contrast to the vulnerability of CNNs against domain shift, the human visual recognition system generalizes incredibly well across domains. For example, young children learn many object concepts from pictures, but they naturally transfer their knowledge to the real world [10].

Similarly, people can easily recognize objects in cartoons or paintings even if they have not seen the same style of an image before. Where does such a difference come from?

A recent line of studies has revealed that standard CNNs have an inductive bias far different from human vision: while humans tend to recognize objects based on their contents (i.e. shapes) [27], CNNs exhibit a strong bias towards styles (i.e. textures) [1, 14, 21]. This may explain why CNNs are intrinsically more sensitive to domain shift because image styles are more likely to change across domains than the contents. Geirhos et al. [14] supported this hypothesis by showing that CNNs trained with heavy augmentation on styles become more robust against various image distortions. Research on CNN architectures [40, 28] has also demonstrated that adjusting the style information in CNNs helps to address multi-domain tasks.

In this paper, we experimentally analyze the relation between CNNs' inductive bias and representation gap across domains, and exploit this relation to address domain shift problems. We propose *Style-Agnostic Networks (SagNets)* which effectively improve CNNs' domain transferability by controlling their inductive bias, without directly reducing domain discrepancy. Our framework consists of separate content-biased and style-biased networks on top of a feature extractor. The content-biased network is encouraged to focus on contents by randomizing styles in a latent space. The style-biased network is led to focus on styles in the opposite way, against which the feature extractor adversarially makes the styles incapable of discriminating class categories. At test time, the prediction is made by the combination of the feature extractor and the content-biased network, where the style bias is substantially reduced.

We show that there exists an apparent correlation between CNNs' inductive bias and their ability to handle domain shift: reducing style bias reduces domain discrepancy. Based on this property, SagNets make significant improvements in a wide range of domain shift scenarios including DG, UDA, and SSDA, across several cross-domain benchmarks such as PACS [29], Office-Home [47], and DomainNet [42].

Our method is orthogonal to the majority of existing domain adaptation and generalization techniques that utilize

\*Equal contribution

<sup>1</sup>Code: <https://github.com/hyeonseobnam/sagnet>

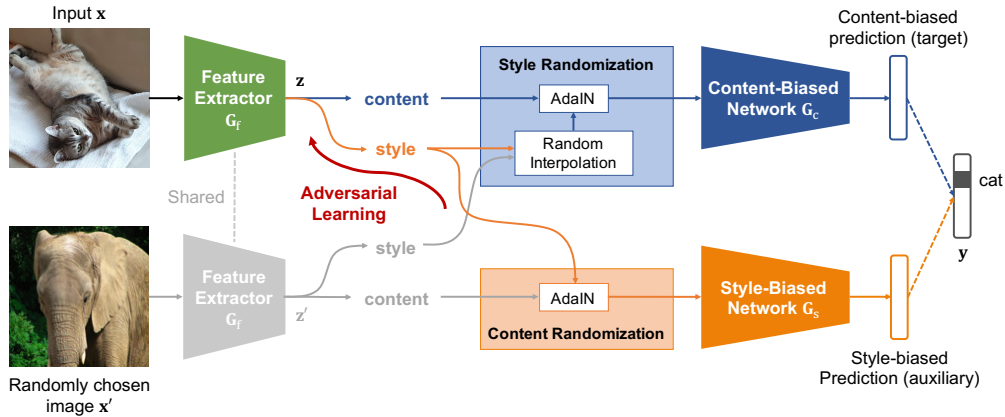


Figure 1: Our Style-Agnostic Network (SagNet) reduces style bias to reduce domain gap. It consists of three sub-networks—a feature extractor, a content-biased network, and a style-biased network—that are jointly trained end-to-end. The content-biased network is led to focus on the content of the input via style randomization (SR), where the style of the input is replaced by an arbitrary style through AdaIN [24]. Conversely, the style-biased network is led to focus on the style by content randomization (CR), while an adversarial learning makes the feature extractor generate less style-biased representation.

domain information for training (e.g. aligning the source and target domains [11, 34, 44]). In other words, SagNets only control the intrinsic bias of CNNs without even requiring domain labels nor multiple domains. This approach is not only scalable to more practical scenarios where domain boundaries are unknown or ambiguous, but also able to complement existing methods and bring additional performance boosts as demonstrated in our extensive experiments.

## 2. Related Work

**Inductive biases of CNNs.** Our work is motivated by the recent findings that CNNs tend to learn styles rather than contents. Geirhos et al. [14] observed that standard ImageNet-trained CNNs are likely to make a style-biased decision on ambiguous stimuli (e.g. images stylized to different categories). Some studies have also shown that CNNs perform well when only local textures are given while global shape structures are missing [12, 6], but work poorly in the reverse scenario [3, 14]. Others have attempted to mitigate the style bias of CNNs under the assumption that it deteriorates CNNs’ generalization capability [23, 14, 21]. However, the practical impact of CNNs’ bias on domain shift problems remains unclear, which we aim to explore in this paper.

**Style Manipulation.** We utilize convolutional feature statistics to control the style bias of CNNs. This owes to previous works dealing with the feature statistics in CNNs, mostly in generative frameworks to manipulate image styles. Gatys et al. [12] showed that feature statistics of a CNN effectively capture the style information of an image, which paved the way for neural style transfer [13, 25, 24]. In particular,

adaptive instance normalization (AdaIN) [24] demonstrated that the style of an image can easily be changed by adjusting the mean and variance of convolutional feature maps. StyleGAN [26] also produced impressive image generation results by repeatedly applying AdaIN operations in a generative network. Manipulating styles has also benefited discriminative problems. BIN [40] improved classification performance by reducing unnecessary style information using trainable normalization, and SRM [28] extended this idea to style-based feature recalibration.

**Domain Generalization and Adaptation.** Domain Generalization (DG) aims to make CNNs robust against novel domains outside the training distribution. A popular approach is learning a shared feature space across multiple source domains, for example, by minimizing Maximum Mean Discrepancy (MMD) [38, 16] or adversarial feature alignment [31, 32]. Some studies divide the model into domain-specific and domain-invariant components using low-rank parameterization [29] or layer aggregation modules [9], assuming that the domain-invariant part well generalizes to other domains. Meta-learning frameworks [2, 8] split the source domains into meta-train and meta-test domains to simulate domain shift, and JiGen [4] employed self-supervised signals such as solving jigsaw puzzles to improve generalization by learning image regularities.

Unsupervised Domain Adaptation (UDA) tackles the problem of domain shift where unlabeled data from the target domain are available for training. The mainstream approach is aligning the source and the target distributions by minimizing MMD [17, 35], adversarial learning [11, 46, 34], or image-level translation [22, 39]. Another problem setting

with domain shift is Semi-Supervised Domain Adaptation (SSDA) where a few target labels are additionally provided. It is commonly addressed by simultaneously minimizing the distance between the domains as well as imposing a regularization constraint on the target data to prevent overfitting [7, 50]. Recently, MME [44] optimized a minimax loss on the conditional entropy of unlabeled data and achieved large performance improvement.

### 3. Style-Agnostic Networks

We present a general framework to make the network agnostic to styles but focused more on contents in an end-to-end manner (Fig. 1). It contains three networks: a feature extractor, a content-biased network, and a style-biased network. The content-biased network is encouraged to exploit image contents when making decisions by randomizing intermediate styles, which we call *content-biased learning*. Conversely, the style-biased network is trained to be biased towards styles by randomizing contents, but it adversarially makes the feature extractor less style-biased, which is referred to as *adversarial style-biased learning*. The final prediction at test time is made by the feature extractor followed by the content-biased network. Following the common practice [13, 25, 24, 26, 40, 28], we utilize the summary statistics of CNN features (i.e. channel-wise mean and standard deviation) as style representation and their spatial configuration as content representation.

#### 3.1. Content-Biased Learning

In content-biased learning, we enforce the model to learn content-biased features by introducing a style randomization (SR) module. It randomizes the styles during training by interpolating the feature statistics between different examples regardless of their class categories. Consequently, the network is encouraged to be invariant to styles and biased toward contents in predicting the class labels.

Given an input training image  $\mathbf{x}$  and a randomly selected image  $\mathbf{x}'$ , we first extract their intermediate feature maps  $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{D \times H \times W}$  from the feature extractor  $\mathbf{G}_f$ , where  $H$  and  $W$  indicate spatial dimensions, and  $D$  is the number of channels. Then we compute the channel-wise mean and standard deviation  $\mu(\mathbf{z}), \sigma(\mathbf{z}) \in \mathbb{R}^D$  as style representation:

$$\mu(\mathbf{z}) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \mathbf{z}_{hw}, \quad (1)$$

$$\sigma(\mathbf{z}) = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (\mathbf{z}_{hw} - \mu(\mathbf{z}))^2 + \epsilon}. \quad (2)$$

The SR module constructs a randomized style  $\hat{\mu}, \hat{\sigma} \in \mathbb{R}^D$  by interpolating between the styles of  $\mathbf{z}$  and  $\mathbf{z}'$ . Then it replaces the style of the input with the randomized style through

adaptive instance normalization (AdaIN) [24]:

$$\hat{\mu} = \alpha \cdot \mu(\mathbf{z}) + (1 - \alpha) \cdot \mu(\mathbf{z}'), \quad (3)$$

$$\hat{\sigma} = \alpha \cdot \sigma(\mathbf{z}) + (1 - \alpha) \cdot \sigma(\mathbf{z}'), \quad (4)$$

$$\text{SR}(\mathbf{z}, \mathbf{z}') = \hat{\sigma} \cdot \left( \frac{\mathbf{z} - \mu(\mathbf{z})}{\sigma(\mathbf{z})} \right) + \hat{\mu}, \quad (5)$$

where  $\alpha \sim \text{Uniform}(0, 1)$  is a random interpolation weight. The style-randomized representation  $\text{SR}(\mathbf{z}, \mathbf{z}')$  is fed into the content-biased network  $\mathbf{G}_c$  to obtain a content-biased loss  $L_c$ , which jointly optimizes the feature extractor  $\mathbf{G}_f$  and the content-biased network  $\mathbf{G}_c$ :

$$\min_{\mathbf{G}_f, \mathbf{G}_c} L_c = -\mathbb{E}_{(\mathbf{x}, \mathbf{y}) \in S} \sum_{k=1}^K \mathbf{y}_k \log \mathbf{G}_c(\text{SR}(\mathbf{G}_f(\mathbf{x}), \mathbf{z}'))_k, \quad (6)$$

where  $K$  is the number of class categories,  $\mathbf{y} \in \{0, 1\}^K$  is the one-hot label for input  $\mathbf{x}$  and  $S$  is the training set.

By randomizing styles during training, the content-biased network can no longer rely on the style but focus more on the content in making a decision. At test time, we remove the SR module and directly connect  $\mathbf{G}_f$  to  $\mathbf{G}_c$ , which ensures independent predictions as well as saves computations.

#### 3.2. Adversarial Style-Biased Learning

In addition to the content-biased learning, we constrain the feature extractor from learning style-biased representation by adopting an adversarial learning framework. In other words, we make the styles encoded by the feature extractor incapable of discriminating the class categories. To achieve this, we build an auxiliary style-biased network  $\mathbf{G}_s$  as a discriminator to make style-biased predictions utilizing a content randomization (CR) module. Then we let the feature extractor  $\mathbf{G}_f$  fool  $\mathbf{G}_s$  via an adversarial framework, which we name adversarial style-biased learning.

Contrary to SR which leaves the content of the input and randomizes its style, the CR module does the opposite: it maintains the style and switches the content to that of an arbitrary image. Given intermediate feature maps  $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{D \times H \times W}$  corresponding to the input  $\mathbf{x}$  and the randomly chosen image  $\mathbf{x}'$ , we apply AdaIN to the content of  $\mathbf{z}'$  with the style of  $\mathbf{z}$ :

$$\text{CR}(\mathbf{z}, \mathbf{z}') = \sigma(\mathbf{z}) \cdot \left( \frac{\mathbf{z}' - \mu(\mathbf{z}')}{\sigma(\mathbf{z}')} \right) + \mu(\mathbf{z}), \quad (7)$$

which can be interpreted as a content-randomized representation of  $\mathbf{z}$ . It is taken as an input to the style-biased network  $\mathbf{G}_s$  which is trained to make a style-biased prediction by minimizing a style-biased loss  $L_s$ :

$$\min_{\mathbf{G}_s} L_s = -\mathbb{E}_{(\mathbf{x}, \mathbf{y}) \in S} \sum_{k=1}^K \mathbf{y}_k \log \mathbf{G}_s(\text{CR}(\mathbf{G}_f(\mathbf{x}), \mathbf{z}'))_k. \quad (8)$$

---

**Algorithm 1: Optimization Process of SagNets**

---

**Input:** training data  $S = (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^M$ ; batch size  $N$ ;  
hyperparameter  $\lambda_{\text{adv}} > 0$   
**Initialize:** feature extractor  $\mathbf{G}_f$ ; content-biased  
network  $\mathbf{G}_c$ ; style-biased network  $\mathbf{G}_s$   
**while not converged do**  
   $\mathbf{X}, \mathbf{Y} = \text{SAMPLEBATCH}(S, N)$   
   $\mathbf{Z} = \mathbf{G}_f(\mathbf{X})$   
   $\mathbf{Z}' = \text{SHUFFLE}(\mathbf{Z})$  // randomly shuffle  
   $\mathbf{Z}$  along the batch dimension.  
  **Content-Biased Learning:**  
   $\mathbf{Z}^{(c)} = \text{SR}(\mathbf{Z}, \mathbf{Z}')$   
   $L_c = -\frac{1}{N} \sum_{j=1}^N \sum_{k=1}^K \mathbf{Y}_{jk} \log \mathbf{G}_c(\mathbf{Z}_j^{(c)})_k$   
  Minimize  $L_c$  w.r.t.  $\mathbf{G}_f$  and  $\mathbf{G}_c$   
  **Adversarial Style-Biased Learning:**  
   $\mathbf{Z}^{(s)} = \text{CR}(\mathbf{Z}, \mathbf{Z}')$   
   $L_s = -\frac{1}{N} \sum_{j=1}^N \sum_{k=1}^K \mathbf{Y}_{jk} \log \mathbf{G}_s(\mathbf{Z}_j^{(s)})_k$   
  Minimize  $L_c$  w.r.t.  $\mathbf{G}_s$   
   $L_{\text{adv}} = -\lambda_{\text{adv}} \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^K \frac{1}{K} \log \mathbf{G}_s(\mathbf{Z}_j^{(s)})_k$   
  Minimize  $L_{\text{adv}}$  w.r.t.  $\mathbf{G}_f$   
**end**  
**Output:**  $\mathbf{G}_c \circ \mathbf{G}_f$

---

The feature extractor  $\mathbf{G}_f$  is then trained to fool  $\mathbf{G}_s$  by minimizing an adversarial loss  $L_{\text{adv}}$  computed by the cross-entropy between the style-biased prediction and uniform distribution<sup>2</sup>:

$$\min_{\mathbf{G}_f} L_{\text{adv}} = -\lambda_{\text{adv}} \mathbb{E}_{(\mathbf{x}, \cdot) \in S} \sum_{k=1}^K \frac{1}{K} \log \mathbf{G}_s(\text{CR}(\mathbf{G}_f(\mathbf{x}), \mathbf{z}'))_k, \quad (9)$$

where  $\lambda_{\text{adv}}$  is a weight coefficient.

Following previous work that utilizes affine transformation parameters (i.e. normalization parameters) to effectively manipulate the style representation [24, 26, 40], we perform the adversarial learning with respect to the affine transformation parameters of  $\mathbf{G}_f$ . We can also efficiently control the trade-off between the content and style biases by adjusting the coefficient  $\lambda_{\text{adv}}$ , which is explored in Sec. 4.1.

### 3.3. Implementation Details

Our framework can be readily integrated into modern CNN architectures and trained end-to-end. Given a CNN such as a ResNet [19], the feature extractor  $\mathbf{G}_f$  comprises first few stages<sup>3</sup> of the CNN, while the rest of the network

<sup>2</sup>This can also be interpreted as maximizing the entropy of style-biased predictions. We empirically found that matching with a uniform distribution provides more stable convergence than directly maximizing the entropy or using a gradient reversal layer [11].

<sup>3</sup>A stage refers to a group of layers that share the same feature map size.

becomes the content-biased network  $\mathbf{G}_c$ ; the style-biased network  $\mathbf{G}_s$  forms the same structure as  $\mathbf{G}_c$ . Consequently, the output network  $\mathbf{G}_c \circ \mathbf{G}_f$  has exactly the same architecture as the original CNN—in other words, it does not impose any overhead in terms of both parameters and computations at test time. To minimize the computational overhead during training, we choose the arbitrary example within a minibatch: given intermediate feature maps  $\mathbf{Z} \in \mathbb{R}^{N \times D \times H \times W}$  from a minibatch of size  $N$ , we construct new feature maps  $\mathbf{Z}'$  corresponding to the arbitrary examples by randomly shuffling  $\mathbf{Z}$  along the batch dimension. The overall training procedure of SagNets is summarized in Algorithm 1.

### 3.4. Extension to Un/Semi-Supervised Learning

Although our framework does not require training data from the target domain, some problem settings such as un/semi-supervised domain adaptation allow access to unlabeled target data for training. To fully leverage those unlabeled data, we utilize a simple extension of SagNets based on consistency learning [37, 49]. For each training example  $\mathbf{x}$  from a set of unlabeled data  $S_{\text{unl}}$ , we obtain two prediction vectors from the network: one applied SR and the other not applied SR. The inconsistency between the two predictions are estimated by the mean square error to form an unsupervised loss  $L_{\text{unl}}$  and minimized for all unlabeled data:

$$\min_{\mathbf{G}_f, \mathbf{G}_c} L_{\text{unl}} = \lambda_{\text{unl}} \mathbb{E}_{\mathbf{x} \in S_{\text{unl}}} \sum_{k=1}^K \{ \mathbf{G}_c(\text{SR}(\mathbf{G}_f(\mathbf{x}), \mathbf{z}'))_k - \mathbf{G}_c(\mathbf{G}_f(\mathbf{x}))_k \}^2, \quad (10)$$

where  $\lambda_{\text{unl}}$  is the corresponding coefficient which is set to 0.01. Furthermore, our adversarial learning is naturally extended to unlabeled data because the adversarial loss (Eq. 9) does not require ground-truth labels.

## 4. Experiments

In this section, we first conduct an experimental analysis to gain an insight into the effect of SagNets (Sec. 4.1), then perform comprehensive evaluation on a wide range of cross-domain tasks including domain generalization (DG) (Sec. 4.2), unsupervised domain adaptation (UDA) (Sec. 4.3), and semi-supervised domain adaptation (SSDA) (Sec. 4.4) compared with existing methods. All networks are pretrained on ImageNet classification [43].

### 4.1. Biases and Domain Gap

We examine the effect of SagNets on CNNs' inductive biases and domain discrepancy using **16-class-ImageNet** [15] and the **texture-shape cue conflict stimuli** [14]. 16-class-ImageNet is a subset of ImageNet containing 213,555 images from 16 entry-level categories. The texture-shape cue conflict stimuli were introduced to quantify the intrinsic biases



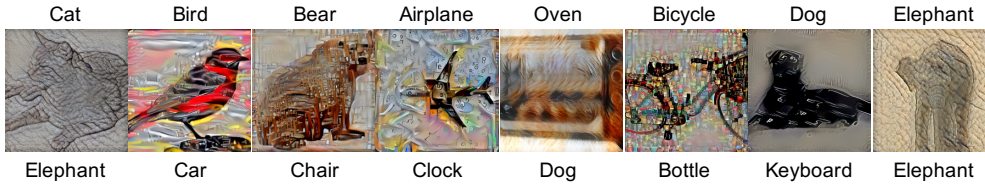


Figure 2: Examples of the texture-shape cue conflict stimuli [14] created by randomly combining the shape (top labels) and texture (bottom labels) of different images through style transfer [13].

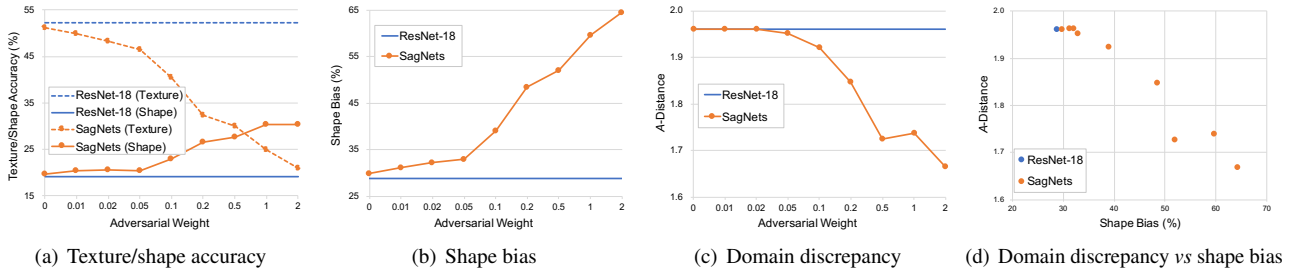


Figure 3: The effect of SagNets on (a) texture/shape accuracy and (b) shape bias on the cue conflict stimuli. The domain discrepancy ( $\mathcal{A}$ -distance) between 16-class-ImageNet and the cue conflict stimuli against (c) the adversarial weight and (d) the shape bias. SagNets significantly increase the shape accuracy and bias, while reducing the domain discrepancy.

of CNNs, which consist of 1,280 images and share the same 16 categories as 16-class-ImageNet. They are generated by blending the texture (style) and shape (content) from different images via style transfer [13] (see examples in Fig. 2), so that we can observe whether a CNN makes a decision based on the texture or shape.

We train SagNets with a ResNet-18 baseline on 16-class-ImageNet only, using SGD with batch size 256, momentum 0.9, weight decay 0.0001, initial learning rate 0.001, and cosine learning rate scheduling for 30 epochs. The randomization stage (the stage after which the randomizations are performed. i.e. the number of stages in the feature extractor) is set to 2, and we vary the adversarial coefficient  $\lambda_{adv}$ .

**Texture/Shape Bias.** As proposed in [14], we quantify the texture and shape biases of networks by evaluating them on the cue conflict stimuli and counting the number of predictions that correctly classify the texture or shape of images. Specifically, shape bias is defined as the fraction of predictions matching the shape within the predictions matching either the shape or texture, and texture bias is defined in a similar way. Fig. 3(a) shows the texture/shape accuracy on the stimuli of the ResNet-18 baseline and SagNets with varying adversarial weight  $\lambda_{adv}$ , which clearly demonstrates that SagNets increase the shape accuracy and decrease the texture accuracy. This consequently increases the shape bias as shown in Fig. 3(b), which is also equivalent to decreasing the texture bias. Furthermore, the shape and texture biases

are effectively controlled by the adversarial weight  $\lambda_{adv}$ , i.e. the shape bias increases as  $\lambda_{adv}$  increases. In practice, increasing  $\lambda_{adv}$  does not always improve the final accuracy because it makes the optimization more difficult, thus we need to find a fair trade-off, which we investigate in Sec 4.2.

**Domain Gap.** We further investigate the capability of SagNets in reducing domain discrepancy. We treat 16-class-ImageNet and the cue conflict stimuli dataset as two different domains because they share the same object categories but exhibit different appearances. We then measure the distance between the two domains using the features from the penultimate layer of the network. Following [33], we calculate a proxy  $\mathcal{A}$ -distance  $d_{\mathcal{A}} = 2(1 - \epsilon)$  where  $\epsilon$  is a generalization error of an SVM classifier trained to distinguish the examples from the two domains. As illustrated in Fig. 3(c), SagNets effectively reduce the domain discrepancy as the adversarial weight increases. By plotting the  $\mathcal{A}$ -distance against the shape bias as presented in Fig. 3(d), we observe an explicit correlation between the bias and domain gap: shape-biased representation generalizes better across domains, which confirms the common intuition [23, 14, 21].

## 4.2. Domain Generalization

DG is a problem to train a model on a single or multiple source domain(s), and test on an unseen target domain. We evaluate the efficacy of SagNets against recent DG methods including D-SAM [9], JiGen [4], Epi-FCR [30], MASF [8],

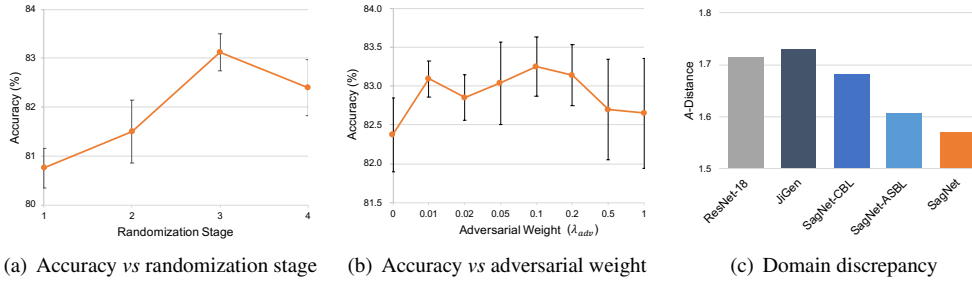


Figure 4: Accuracy of SagNets on PACS with varying (a) randomization stage and (b) adversarial weight; (c) the domain discrepancy ( $\mathcal{A}$ -distance) between the source domains and the target domain. Results are averaged over the 4 target domains and 3 repetitions, where error bars denote the standard deviation.

Table 1: Multi-source domain generalization accuracy (%) on PACS. Each column title indicates the target domain. The results of our DeepAll baselines (standard supervised learning on the mixture of source domains) and SagNets are averaged over three repetitions.  $\text{SagNet}^{-\text{CBL}}$  and  $\text{SagNet}^{-\text{ASBL}}$  refer to SagNets without content-biased learning and adversarial style-biased learning, respectively. Our approach outperforms all competing methods, where each component of our method contributes to the performance improvements.

	Art paint.	Cartoon	Sketch	Photo	Avg.
AlexNet					
D-SAM	63.87	70.70	64.66	85.55	71.20
JiGen	67.63	71.71	65.18	89.00	73.38
Epi-FCR	64.7	72.3	65.0	86.1	72.0
MASF	70.35	72.46	67.33	<b>90.68</b>	75.21
MMLD	69.27	<b>72.83</b>	66.44	88.98	74.38
DeepAll	65.19	67.83	63.75	90.08	71.71
SagNet	<b>71.01</b>	70.78	<b>70.26</b>	90.04	<b>75.52</b>
ResNet-18					
D-SAM	77.33	72.43	<b>77.83</b>	95.30	80.72
JiGen	79.42	75.25	71.35	96.03	80.51
Epi-FCR	82.1	77.0	73.0	93.9	81.5
MASF	80.29	77.17	71.69	94.99	81.04
MMLD	81.28	77.16	72.29	<b>96.09</b>	81.83
DeepAll	78.12	75.10	68.43	95.37	79.26
$\text{SagNet}^{-\text{CBL}}$	78.86	77.05	73.28	95.43	81.15
$\text{SagNet}^{-\text{ASBL}}$	82.94	76.73	74.74	95.07	82.37
SagNet	<b>83.58</b>	<b>77.66</b>	76.30	95.47	<b>83.25</b>

Table 2: Multi-source domain generalization accuracy (%) on Office-Home with a ResNet-18 backbone.

	Art	Clipart	Product	Real-World	Avg.
D-SAM	58.03	44.37	69.22	71.45	60.77
JiGen	53.04	<b>47.51</b>	<b>71.47</b>	72.79	61.20
DeepAll	58.51	41.44	70.06	73.28	60.82
$\text{SagNet}^{-\text{CBL}}$	60.00	42.85	70.11	73.12	61.52
$\text{SagNet}^{-\text{ASBL}}$	59.31	41.89	70.44	<b>73.52</b>	61.29
SagNet	<b>60.20</b>	45.38	70.42	73.38	<b>62.34</b>

and MMLD [36]. We adopt two multi-domain datasets: **PACS** [29] consists of 9,991 images from 7 categories across 4 domains (Art Painting, Cartoon, Sketch, and Photo) and **Office-Home** [47] comprises 15,588 images from 65 categories and 4 domains (Art, Clipart, Product and Real-World). We split the training data of PACS into 70% training and 30% validation following the official split [29], and Office-Home into 90% training and 10% validation following [9]. Our networks are trained by SGD with batch size 96, momentum 0.9, weight decay 0.0001, initial learning rate 0.004 (0.002 with AlexNet) and cosine scheduling for 2K iterations (4K with AlexNet or Office-Home). The randomization stage and the adversarial weight of SagNets are fixed to 3 and 0.1, respectively, throughout all remaining experiments unless otherwise specified.

**Multi-Source Domain Generalization.** We first examine multi-source DG where the model needs to generalize from multiple source domains to a novel target domain. We train our SagNets and the DeepAll baselines (i.e. naïve supervised learning) on the combination of all training data from the source domains regardless of their domain labels. Table 1 and 2 demonstrate that SagNets not only significantly improve the accuracy over the DeepAll baselines but also outperform the competing methods. Furthermore, experiments without content-biased learning ( $\text{SagNet}^{-\text{CBL}}$ ) and adversarial style-biases learning ( $\text{SagNet}^{-\text{ASBL}}$ ) verify the effectiveness of each component of SagNets. It is also worth noting that while all the compared methods except JiGen exploit additional layers on top of the baseline CNN at test time, SagNets do not require any extra parameters nor computations over the baseline.

**Ablation.** We perform an extensive ablation study for SagNets on multi-source DG using PACS. We first examine the effect of the randomization stage where the randomizations are applied, while keeping the adversarial weight to 0.1. Karas et al. [26] demonstrated that styles at different layers

Table 3: Single-source domain generalization accuracy (%) on PACS averaged over three repetitions (A: Art Painting, C: Cartoon, S: Sketch, P: Photo).

	A→C	A→S	A→P	C→A	C→S	C→P	S→A	S→C	S→P	P→A	P→C	P→S	Avg.
ResNet-18	62.3	49.0	95.2	65.7	60.7	83.6	28.0	54.5	35.6	64.1	23.6	29.1	54.3
JiGen	57.0	50.0	<b>96.1</b>	65.3	65.9	85.5	26.6	41.1	42.8	62.4	27.2	35.5	54.6
ADA	64.3	<b>58.5</b>	94.5	66.7	65.6	83.6	37.0	58.6	41.6	65.3	32.7	35.9	58.7
SagNet	<b>67.1</b>	56.8	95.7	<b>72.1</b>	<b>69.2</b>	<b>85.7</b>	<b>41.1</b>	<b>62.9</b>	<b>46.2</b>	<b>69.8</b>	<b>35.1</b>	<b>40.7</b>	<b>61.9</b>

Table 4: Unsupervised domain adaptation accuracy (%) on Office-Home with varying adaptation methods and their SagNet combinations (A: Art, C: Clipart, P: Product, R: Real-World). SagNets consistently boost the performance when combined with various adaptation methods.

Method	SagNet	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
ResNet-50		41.3	63.8	71.4	49.1	59.6	61.4	46.8	36.1	68.8	63.0	45.9	76.5	57.0
	✓	<b>45.7</b>	<b>64.1</b>	<b>72.6</b>	<b>49.6</b>	<b>60.0</b>	<b>63.5</b>	<b>49.9</b>	<b>40.7</b>	<b>71.1</b>	<b>64.8</b>	<b>50.9</b>	<b>78.1</b>	<b>59.2</b>
DANN		44.7	62.7	70.3	47.1	60.1	61.4	46.1	41.7	68.5	62.3	50.9	76.7	57.7
	✓	<b>48.8</b>	<b>65.2</b>	<b>71.4</b>	<b>50.3</b>	<b>61.4</b>	<b>62.5</b>	<b>50.7</b>	<b>45.7</b>	<b>71.8</b>	<b>65.4</b>	<b>55.2</b>	<b>78.6</b>	<b>60.6</b>
JAN		45.0	63.3	72.6	53.3	<b>66.0</b>	64.4	50.9	40.8	72.1	64.9	49.4	78.8	60.1
	✓	<b>50.1</b>	<b>66.8</b>	<b>73.9</b>	<b>56.9</b>	64.7	<b>66.1</b>	<b>54.9</b>	<b>45.6</b>	<b>75.2</b>	<b>70.0</b>	<b>55.3</b>	<b>80.1</b>	<b>63.3</b>
CDAN		50.6	69.0	<b>74.9</b>	54.6	66.1	67.9	57.2	46.9	75.6	69.1	55.8	<b>80.6</b>	64.0
	✓	<b>53.2</b>	<b>69.2</b>	<b>74.9</b>	<b>55.9</b>	<b>67.8</b>	<b>68.6</b>	<b>58.1</b>	<b>51.8</b>	<b>76.4</b>	<b>69.8</b>	<b>58.1</b>	80.4	<b>65.3</b>
SymNet		47.2	70.9	77.4	64.5	<b>71.7</b>	73.0	63.3	49.0	<b>79.1</b>	74.2	54.1	82.7	67.3
	✓	<b>49.6</b>	<b>72.4</b>	<b>77.9</b>	<b>63.9</b>	71.1	<b>72.5</b>	<b>65.2</b>	<b>51.0</b>	<b>79.1</b>	<b>74.9</b>	<b>56.1</b>	<b>83.0</b>	<b>68.0</b>

encode distinct visual attributes: the styles from fine spatial resolutions (lower layers in our network) encode low-level attributes such as color and microtextures, while the styles from coarse spatial resolutions (higher layers in our network) encode high-level attributes such as global structures and macrotextures. In this regard, the randomization modules of SagNets need to be applied at a proper level, where the style incurs undesired bias. As shown in Fig. 4(a), SagNets offer the best improvement when the randomizations are applied after stage 3, while randomizing too low-level styles is less helpful in reducing style bias; randomizing too high-level styles may lose important semantic information.

We also conduct ablation on the adversarial coefficient  $\lambda_{adv}$ , while the randomization stage is fixed to 3. Fig. 4(b) illustrates the accuracy of SagNets with varying values of  $\lambda_{adv}$ . Although increasing  $\lambda_{adv}$  tends to steadily improve the shape bias as shown in Fig. 3(b), the actual performance peaks around  $\lambda_{adv} = 0.1$ . This indicates that while increasing the shape bias leads to reducing domain discrepancy, it may complicate the optimization and damage the performance when exceeding a proper range.

**Domain Gap.** Reducing domain gap in DG setting is particularly challenging compared with other tasks such as UDA or SSDA since one can not make any use of the target distribution. To demonstrate the effect of SagNets in such scenario, we measure the  $\mathcal{A}$ -distances between the source domains and the target domain following the same procedure as in Sec. 4.1, and average them over the 4 generalization

tasks of PACS. As shown in Fig. 4(c), SagNets considerably reduce the domain discrepancy compared to the ResNet-18 baseline, JiGen<sup>4</sup>, and ablated SagNets.

**Single-Source Domain Generalization.** Our framework seamlessly extends to single-source DG where only a single training domain is provided, because it does not require domain labels nor multiple source domains (which are necessary in the majority of DG methods [29, 9, 30, 8]). We train SagNets on each domain of PACS and evaluate them on the remaining domains. As reported in Table 3, SagNets remarkably boost the generalization performance, while JiGen and ADA<sup>5</sup> [48] are technically applicable to single-source DG but outperformed by our method.

### 4.3. Unsupervised Domain Adaptation

UDA is a task of transferring knowledge from a source to the target domain, where unlabeled target data are available for training. Besides **Office-Home** that we used for DG, we also employ **DomainNet** [42] which is a large-scale dataset containing about 0.6 million images from 6 domains and 345 categories. Since some domains and classes are very noisy, we follow [44] to utilize its subset of 145,145 images from

<sup>4</sup>We provide additional comparison with JiGen because it does not use domain labels as same as ours. The results are reproduced with their official code (<https://github.com/fmcarlucci/JigenDG>) and the optimal hyperparameters provided in their paper.

<sup>5</sup>We re-implement their method on top of our baseline, which gives better performance than reproduction based on their published code.

Table 5: Unsupervised domain adaptation accuracy (%) on DomainNet (C: Clipart, P: Painting, S: Sketch, R: Real). SagNets improve the performance over both ResNet-18 and CDAN [34] baselines.

Method	SagNet	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Avg.
ResNet-18	✓	53.1	57.7	52.4	47.5	52.0	43.4	<b>68.5</b>	53.5
		<b>54.4</b>	<b>58.0</b>	<b>53.1</b>	<b>49.2</b>	<b>52.2</b>	<b>46.4</b>	67.4	<b>54.4</b>
CDAN	✓	53.0	57.4	52.3	48.0	52.3	43.4	<b>67.2</b>	53.4
		<b>54.4</b>	<b>59.4</b>	<b>52.8</b>	<b>49.5</b>	<b>52.4</b>	<b>45.9</b>	67.1	<b>54.5</b>

Table 6: Semi-supervised domain adaptation accuracy (%) on DomainNet (C: Clipart, P: Painting, S: Sketch, R: Real). SagNets consistently improve the performance over various baselines in terms of backbone architectures and adaptation methods.

Backbone	Method	SagNet	R → C		R → P		P → C		C → S		S → P		R → S		P → R		Avg.	
			1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
AlexNet	S+T	✓	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	<b>55.8</b>	<b>58.7</b>	40.0	43.4
	MME	✓	<b>45.8</b>	<b>49.1</b>	<b>45.6</b>	<b>46.7</b>	<b>42.7</b>	<b>46.3</b>	<b>36.1</b>	<b>39.4</b>	<b>37.1</b>	<b>39.8</b>	<b>34.2</b>	<b>37.5</b>	54.0	57.0	<b>42.2</b>	<b>45.1</b>
VGG-16	S+T	✓	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	<b>60.7</b>	44.2	48.2
	MME	✓	<b>54.1</b>	<b>58.6</b>	<b>49.8</b>	<b>52.2</b>	<b>48.7</b>	<b>54.4</b>	<b>39.6</b>	<b>43.4</b>	<b>40.4</b>	<b>43.4</b>	<b>39.7</b>	<b>42.8</b>	<b>57.0</b>	<b>60.3</b>	<b>47.0</b>	<b>50.7</b>
ResNet-34	S+T	✓	49.0	52.3	55.4	56.7	47.7	51.0	43.9	48.5	50.8	55.1	37.9	45.0	<b>69.0</b>	<b>71.7</b>	50.5	54.3
	MME	✓	<b>51.8</b>	<b>54.9</b>	<b>57.8</b>	<b>59.4</b>	<b>50.4</b>	<b>54.2</b>	<b>48.9</b>	<b>52.9</b>	<b>53.1</b>	<b>56.3</b>	<b>45.6</b>	<b>49.4</b>	68.3	70.9	<b>53.7</b>	<b>56.9</b>
ResNet-18	S+T	✓	60.6	64.1	63.3	63.5	57.0	60.7	50.9	55.4	<b>60.5</b>	60.9	50.2	54.8	<b>72.2</b>	<b>75.3</b>	59.2	62.1
	MME	✓	<b>64.9</b>	<b>67.8</b>	<b>64.5</b>	<b>66.0</b>	<b>60.4</b>	<b>65.8</b>	<b>54.7</b>	<b>59.0</b>	59.8	<b>62.0</b>	<b>56.6</b>	<b>59.6</b>	71.1	74.2	<b>61.7</b>	<b>64.9</b>
ResNet-34	S+T	✓	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	<b>59.5</b>	46.3	50.1	71.8	<b>73.9</b>	56.9	60.0
	MME	✓	<b>59.4</b>	<b>62.0</b>	<b>61.9</b>	<b>62.9</b>	<b>59.1</b>	<b>61.5</b>	<b>54.0</b>	<b>57.1</b>	<b>56.6</b>	59.0	<b>49.7</b>	<b>54.4</b>	<b>72.2</b>	73.4	<b>59.0</b>	<b>61.5</b>
ResNet-18	S+T	✓	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	<b>76.1</b>	<b>78.5</b>	66.4	68.9
	MME	✓	<b>72.3</b>	<b>74.2</b>	<b>69.0</b>	<b>70.5</b>	<b>70.8</b>	<b>73.2</b>	<b>61.7</b>	<b>64.6</b>	<b>66.9</b>	<b>68.3</b>	<b>64.3</b>	<b>66.1</b>	75.3	78.4	<b>68.6</b>	<b>70.8</b>

4 domains (Clipart, Painting, Sketch, and Real) with 126 categories, and consider 7 adaptation scenarios. We demonstrate that SagNets are not only effective when used alone, but also able to complement popular UDA methods such as DANN [11], JAN [35], CDAN [34], and SymNet [51] to make further improvements. We reproduce all compared methods and their SagNet variants on top of the same baseline and training policy for fair comparison.

As shown in Table 4, SagNets lead to impressive performance improvements when combined with any adaptation method. Furthermore, while CDAN does not make meaningful improvement on DomainNet due to the complexity of the dataset in terms of scale and diversity, SagNets consistently improve the accuracy in most adaptation scenarios as presented in Table 5. These results indicate that the effectiveness of SagNets is orthogonal to existing approaches and generates a synergy by complementing them.

#### 4.4. Semi-Supervised Domain Adaptation

We also conduct SSDA on **DomainNet** where a few target labels are provided. Saito et al. [44] showed that common UDA methods [11, 34, 45] often fail to improve their performance in such scenario, and proposed minimax entropy (MME) setting the state-of-the-art for SSDA. Here we show that SagNets are successfully applied to SSDA, bringing further improvements over MME. For a fair comparison, SagNets are built upon the official implementation of MME<sup>6</sup>,

<sup>6</sup>[https://github.com/VisionLearningGroup/SSDA\\_MME](https://github.com/VisionLearningGroup/SSDA_MME)

trained using the same training policy and the few-shot labels specified by [44].

Table 6 illustrates the results of SSDA with various architectures, where S+T indicates the baseline few-shot learning algorithm based on cosine similarity learning [5]. Our method consistently improves the performance with considerable margins in most adaptation tasks with respect to all tested architectures and baseline methods, which verifies its scalability to various conditions.

## 5. Conclusion

We present Style-Agnostic Networks (SagNets) that are robust against domain shift caused by style variability across domains. By randomizing styles in a latent feature space and adversarially disentangling styles from class categories, SagNets are trained to rely more on contents rather than styles in their decision making process. Our extensive experiments confirm the effectiveness of SagNets in controlling the inductive biases of CNNs and reducing domain discrepancy in a broad range of problem settings.

Our work is orthogonal to many existing domain adaptation approaches that explicitly align the distributions of different domains, and achieves further improvements by complementing those methods. The principle of how we deal with the intrinsic biases of CNNs can also be applied to other areas such as improving robustness under image corruptions [20] and defending against adversarial attacks [18], which are left for future investigation.



## References

- [1] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLOS Computational Biology*, 2018. 1
- [2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018. 2
- [3] Pedro Ballester and Ricardo Matsumura Araujo. On the performance of googlenet and alexnet applied to sketches. In *AAAI*, 2016. 2
- [4] Fabio Maria Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019. 1, 2, 5
- [5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 8
- [6] Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. Brendel, Wieland and Bethge, Matthias. In *ICLR*, 2019. 2
- [7] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013. 1, 3
- [8] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019. 1, 2, 5, 7
- [9] Antonio D’Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *GCPR*, 2018. 1, 2, 5, 6, 7
- [10] Patricia A Ganea, Megan Bloom Pickard, and Judy S DeLoache. Transfer between picture books and the real world by very young children. *Journal of Cognition and Development*, 2008. 1
- [11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 2, 4, 8
- [12] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015. 2
- [13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2, 3, 5
- [14] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 1, 2, 4, 5
- [15] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *NeurIPS*, 2018. 4
- [16] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *TPAMI*, 2016. 2
- [17] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *PRICAI*, 2014. 2
- [18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 8
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4
- [20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 8
- [21] Katherine L Hermann and Simon Kornblith. Exploring the origins and prevalence of texture bias in convolutional neural networks. *arXiv preprint arXiv:1911.09071*, 2019. 1, 2, 5
- [22] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle consistent adversarial domain adaptation. In *ICML*, 2018. 2
- [23] Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Poovendran. Assessing shape bias property of convolutional neural networks. In *CVPR Workshops*, 2018. 2, 5
- [24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 3, 4
- [25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2, 3
- [26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 3, 4, 6
- [27] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Journal of Cognition and Development*, 1988. 1
- [28] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. In *ICCV*, 2019. 1, 2, 3
- [29] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 1, 2, 6, 7
- [30] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *ICCV*, October 2019. 5, 7
- [31] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 2
- [32] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018. 2
- [33] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1, 5
- [34] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 1, 2, 8

- [35] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017. 2, 8
- [36] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, 2020. 6
- [37] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 2018. 4
- [38] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 2
- [39] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018. 2
- [40] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *NeurIPS*, 2018. 1, 2, 3, 4
- [41] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2009. 1
- [42] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 1, 7
- [43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 4
- [44] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019. 1, 2, 3, 7, 8
- [45] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *ICLR*, 2018. 8
- [46] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1, 2
- [47] Hemant Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 1, 6
- [48] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 2018. 7
- [49] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, 2018. 4
- [50] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015. 1, 3
- [51] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, 2019. 8