Adversarial Bayesian Augmentation for Single-Source Domain Generalization

Sheng Cheng [1] Tejas Gokhale [1,2] Yezhou Yang [1]
[1] Arizona State University [2] University of Maryland, Baltimore County
scheng53@asu.edu, gokhale@umbc.edu, yz.yang@asu.edu

Abstract

Generalizing to unseen image domains is a challenging problem primarily due to the lack of diverse training data, inaccessible target data, and the large domain shift that may exist in many real-world settings. As such data augmentation is a critical component of domain generalization methods that seek to address this problem. We present Adversarial Bayesian Augmentation (ABA), a novel algorithm that learns to generate image augmentations in the challenging single-source domain generalization setting. ABA draws on the strengths of adversarial learning and Bayesian neural networks to guide the generation of diverse data augmentations – these synthesized image domains aid the classifier in generalizing to unseen domains. We demonstrate the strength of ABA on several types of domain shift including style shift, subpopulation shift, and shift in the medical imaging setting. ABA outperforms all previous state-of-the-art methods, including pre-specified augmentations, pixel-based and convolutional-based augmentations. Code: https://github.com/shengcheng/ABA.

1. Introduction

Improving the generalization of deep neural networks to out-of-distribution samples is a fundamental yet challenging problem in machine learning and computer vision [48, 25, 34]. Typically, neural networks are trained and tested on data samples from the same distribution (under the *i.i.d.* assumption); under this setting, image classifiers have achieved impressive performances. However, in real-world applications, the distribution of test samples can drastically differ from the training samples [47, 37]. This is especially problematic when the process of acquiring labeled samples from the target test domain is expensive or infeasible, making it difficult to apply semi-supervised learning for domain adaptation [55, 53]. Therefore, there is a need to develop techniques that enable deep neural networks to capture the domain-invariant patterns in the data [34, 51], facilitating improved generalization to out-of-distribution samples.

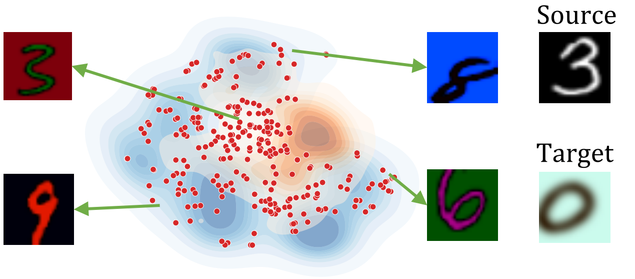

Figure 1: An illustration of the diversity introduced by Adversial Bayesian Augmentations. The blue and orange surfaces represent the source (seen) and target (unseen) domains respectively. The red dots represent the samples augmented by ABA; these augmentations expose the classifier to regions closer to the target domain, thereby improving image classifiers' generalization to unseen domains.

In the multi-source domain generalization (MSDG) setting, where there are multiple source domains for training, domain label information can be leveraged to learn the domain shift [34, 7, 51]. Prior information about the target domain is also useful to design specific data augmentation methods to tackle domain shift. For instance, if it is known that the target domain contains sketches, skeletonizing the source images is a good solution [13]; if it is known that the target domain contains geometric transformations, rotation/translation/scaling would be a suitable augmentation [21]; or if attributes of the target domain are known they can be used for learning data augmentations [14]. However, these methods assume that we know the properties of the target domain – such knowledge is not available in the single-source domain generalization (SSDG) setting. In the SSDG setting, where only one domain is available for training, it is more challenging to address the domain shift issue. In this paper, we focus on the strict SSDG setting, where only one source domain is available for training and no prior knowledge is available about the target domain.

Recent work in SSDG focuses on augmenting the data in order to simulate the presence of out-of-distribution do-

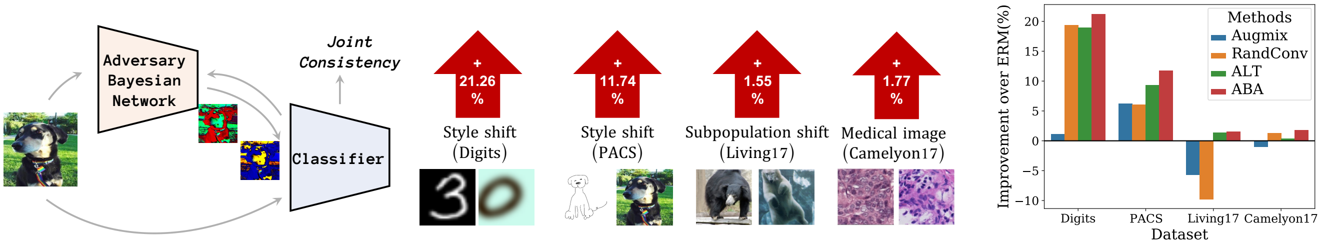

Figure 2: The Adversarial Bayesian Augmentation framework (left panel), the improvement of our method (ABA$_{\text{5-layers}}$) over ERM on each dataset, where samples in source and target domain are displayed under the name of the dataset (middle panel), and summarization of results on a wide range of domain generalization benchmarks (right panel).

mains. One way involves learning-free data augmentation methods, such as RandConv [52], Augmix [19] and JiGen [3] – here the data augmentation is pre-specified and does not evolve or adapt during training. Another approach is based on adversarial perturbations, which involves generating adversarial samples to improve generalization, such as Augmax [49], ADA [47], M-ADA [39], and ALT [15]. Although the Bayesian neural networks as the backbone of the classifier show good generalization ability to out-of-distribution samples intrinsically [30, 51, 4], and some papers [40] use Bayesian neural networks for generating images, none of the work directly augments the data by Bayesian neural network for domain generalization.

In this paper, we present a novel approach called Adversarial Bayesian Augmentation, dubbed *ABA*, which draws on the strengths of adversarial learning and Bayesian neural networks to generate more diverse data and improve generalization on different domains, as shown in Figure 2. Specifically, the adversarial learning-based methods, which explore a wider augmentation space, already outperforms learning-free methods [15, 49] on SSDG. The introduction of weight uncertainty by the Bayesian neural network further enhances the strength of data augmentation, as shown in Figure 1. Our experimental results demonstrate the superior performance of ABA compared to existing methods.

The key contributions and findings of the paper thus are:

- We introduce a novel data augmentation method, dubbed ABA, which combines adversarial learning and Bayesian neural network, to improve the diversity of training data for single-source domain generalization setting.
- We empirically validate the effectiveness of our proposed method on four datasets, covering three types of domain generalization, namely style generalization, subpopulation generalization, and medical imaging generalization. Our method outperforms all existing state-of-art methods on all four datasets.
- We investigate the core driving forces from ABA which enable the generation of diverse data and conduct comprehensive ablation study on how the model hyperparameters impact the overall system's performance.

2. Related Work

Domain Generalization. Domain generalization aims to learn representations that can be transferred to unseen domains. Recent research has explored a range of techniques, including feature fusion [42], meta learning [27], adversarial learning [47, 39], style transformation [35]. Typically, there are two settings of domain generalization: single-source domain generalization (SSDG), where only one source domain is available for training, and multi-source domain generalization (MSDG), where multiple source domains are provided. However, in practice, data from multiple sources can be expensive or sometimes infeasible. In this paper, we focus on SSDG setting.

Single-Source Domain Generalization. Since there is no extra information available about the target domain, most methods for SSDG focus on data augmentation to generate more diverse samples. For example, JiGen [3] decomposes the image into grids and randomly shuffles the patches to create the augmented image. RandConv [52] uses random convolutions as data augmentation, preserving the shape and local texture information. ADA [47] and M-ADA [39] adversarially augment image at the pixel level. SagNet [35] transfers the style of the image. Augmix [19] and Augmax [49] compose data augmentation operations with random or learned mixing coefficients.

Bayesian Neural Network. In a standard deep neural network, the weight of the network takes the single values learned from data. In contrast, the Bayesian neural network (BNN) aims to estimate the distribution of weights, which provides the capability of uncertainty estimation [10], robustness to over-fitting [9], and resistance to adversarial attacks [38]. However, the estimation of the posterior of the weights is often intractable. Current research has focused on techniques such as Bayes By Backprop (BBB) [2] with local reparameterization [22, 33], Variational Inference (VI) [23], and Flipout approximation [50]. Variational Bayesian inference, coupled with domain invariance learning [51] can improve domain generalization. However, this method is only adapted to MSDG settings.

Adversarial Training. Adversarial training has been proposed as a solution to mitigate against the vulnerability of neural networks to input perturbations [44, 17, 31]. AdvBNN [29] proposed an adversarial-trained Bayesian neural network that is robust to strong adversarial attacks, which we adopt with a similar formulation of the min-max problem. Recent work [43] has shown that on-manifold adversarial samples can improve both robustness and generalization. Adversarial training has also been adopted for domain generalization. ADA [47] and M-ADA [39] employ adversarial data augmentation at the pixel level to generate difficult examples for improving domain generalization. ESDA [46] adversarially learns image transformation, while Augmax [49] learns the combination of image augmentation operations in an adversarial manner. ALT [15] goes further and builds an additional image-to-image transformation network to learn adversarial augmentations. Recent work shows trade-offs between reliability metrics such as accuracy, robustness, and fairness [54, 16, 32, 45].

3. Proposed Method

Let $\mathcal{S}$ and $\mathcal{T}$ represent the source and target domains respectively, which share the same label space. The training set is a subset in the source domain and contains N training pairs, denoted as $\{(x_i, y_i)\}_{i=1}^N \subset \mathcal{S}$. The objective of SSDG is to use $\mathcal{S}$ to learn parameters θ of a classifier f which also can generalize well to target domain $\mathcal{T}$.

To accomplish this, since no information is available from the target domain $\mathcal{T}$, previous works focus on data augmentation, denoted as g. For example, in RandConv [52], g is a random convolutional layer, while in Augmix [19], g is a composition of image augmentation operations. To learn the representation invariant to data augmentation, a consistency regularization loss is typically used to encourage consistent prediction between the clean image and the augmented image. The Kullback-Leibler (KL) divergence loss is commonly used for consistency loss.

3.1. Adversarial Bayesian Augmentation

In this paper, we design g as a L-layer Bayesian convolutional neural network, parameterized by $\Phi = \{\phi_l\}_{l=1}^L$, where $\phi_i \in \mathbb{R}^{k_l \times k_l \times C_{in(l)} \times C_{out(l)}}$ are the parameters of each Bayesian convolutional layer. Following the setting in [52], we randomly sample k_l from $\mathcal{K} = \{1, 3, ...n\}$. $C_{in(l)}$ and $C_{out(l)}$ represent the number of input and output channels for each layer convolutional kernel. Since g is an image augmentation function, the number of input channels for the first and last layer are equal to the number of image channels (3 for color images and 1 for grayscale images).

To perform Bayesian inference, we need to estimate the posterior distribution $p(\phi_l|x, y)$, which is intractable in closed form. To approximate it, we adopt the variational Bayesian inference approach and use a variational distribution $q(\phi_l)$. This distribution is obtained by minimizing the KL divergence between $q(\phi_l)$ and true posterior distribution $p(\phi_l|x, y)$. To enable efficient sampling of the variational distribution, we re-parameterize as $\phi_l = \mu_l + \sigma_l \epsilon_l$, where ϵ_l is sampled from the standard normal distribution, which allows us to compute the gradients of μ_l and σ_l. We denote $\boldsymbol{\mu} = \{\mu_l\}_{l=1}^L$ and $\boldsymbol{\sigma} = \{\sigma_l\}_{l=1}^L$. So $\Phi = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$.

Algorithm 1: Learning with Adversarial Bayesian Augmentation (ABA)

```
Input  : $\{x_i, y_i\}_{i=1}^N$
Output : Classifier $f$ parameters $\theta^*$
for $t \leftarrow 1$ to $T$ do
    if $t < T_{warmup}$ then
        $\theta \leftarrow \theta - \gamma \bigtriangledown \mathcal{L}_{\text{cls}}$
    else
        /* Training ABA */
        $\Phi \leftarrow \Phi_0$
        for $m \leftarrow 1$ to $T_{adv}$ do
            $y_g = f(g(x, \Phi), \theta)$
            $\Phi \leftarrow \Phi - \eta \bigtriangledown \mathcal{L}_{\text{ELBO}}$ // See (1)
        end for
        /* Train classifier */
        $\Phi \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$ // Sample parameters
        $\theta \leftarrow \theta - \gamma \bigtriangledown (\mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{KL}})$ // See (2),(3)
    end if
end for
Return $\theta$
```

The optimization of ABA is formulated as a min-max problem. Initially, we optimize the g network using adversarial training to augment images that can fool the classifier f. To achieve this, we use the evidence lower bound (ELBO) of the variational Bayesian network as the loss function. ELBO is a lower bound on the log marginal likelihood of the observed data and is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{ELBO}} = & \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{g \sim q(\Phi)} [\log(y_i | g(x_i), \theta)] \\ & - \beta \sum_{l=1}^{L} \text{KL}(q(\phi_l) || p(\phi_l)), \end{aligned} \tag{1}$$

where the prior distribution $p(\phi_l)$ of each layer follows $\mathcal{N}(0, \frac{1}{k_l \times k_l \times C_{in(l)}})$, which is commonly used in network initialization [18]. Theoretically, the coefficient β for the KL term should be 1. However, in practice, for small datasets or large models, smaller β $(0 < \beta < 1)$ is preferred [29].

Starting from a random initialization, the parameters of g are iteratively updated by maximizing the negative of ELBO. In contrast to adv-BNN [29], which constrains the adversarial samples bounded by ℓ_p norm, we control the strength of adversarial samples by adjusting the learning rate η and the number of adversarial steps T_{adv}. The final augmented images x_g are obtained through Bayesian inference using the optimized parameters Φ^* and clamped to the image range. Note that we can sample multiple augmented images from Bayesian inference, and we sample twice denoted as x_{g_1} and x_{g_2}. These augmented images can be used for classifier learning in the presence of domain shift.

Next, we optimize the classifier f with a loss function consisting of two terms: a cross-entropy loss, which is

$$\mathcal{L}_{\text{cls}} = \text{CrossEntropy}(f(x_{g_1}, \theta), y), \tag{2}$$

and a consistency regularization loss, which helps to keep the prediction consistent on augmented data, defined as:

$$\mathcal{L}_{\text{KL}} = \text{KL}(p_c||\bar{p}) + \text{KL}(p_{g_1}||\bar{p}) + \text{KL}(p_{g_2}||\bar{p}), \tag{3}$$

where p_c, p_{g_1}, p_{g_2} denotes the softmax prediction of f on clean image x and augmented images x_{g_1}, x_{g_2} respectively. $\bar{p}$ is the average of p_c, p_{g_1}, and p_{g_2}.

Comparison with other convolutional-based augmentations. For only one layer network, ABA adversarially learns the distribution of parameters, while in RandConv [52], the parameters are sampled from a fixed distribution, namely $\mathcal{N}(0, \frac{1}{C_{in} \times H \times W})$, where H and W are the size of the convolutional kernel, C_{in} is the input channel. For multiple layers network, compared with ALT [15], ABA learns a *Bayesian* convolutional neural network adversarially rather than a standard convolutional neural network.

Implementation. Algorithm 1 depicts the implementation details. For network design, the activation of multiple layers ABA is `LeakyRelu`. The second augmented image x_{g_2} can be obtained not only through Bayesian inference, but also obtained from other data augmentation techniques, such as RandConv [52], Augmix [19]. We train the classifier for a total of T iterations. At first T_{warmup} iterations, we train the classifier without any data augmentation methods. After T_{warmup}, for each iteration, we randomly initialize the g and update its parameters by adversarial Bayesian training. The learning rate of adversarial learning is η. After T_{adv} steps adversarial learning, we sample the augmented images via Bayesian inference and clamp them to the image range. We then use the augmented images, along with the clean image, to train the classifier f using the classification loss and consistency regularization. The learning rate of the classifier is γ and the weight of consistency regularization is α.

4. Experiments

In this section, we validate our method on four datasets that represent three types of generalization: style shift, subpopulation shift, and domain shift in medical imaging. We train our models on the source domain and evaluate them on the test set of each target domain.

We compare our approach against several state-of-the-art methods [1] using seven variants. For fair comparison with RandConv [52], we use ABA$_{\text{1-layer}}$, *i.e.* ABA with a 1-layer Bayesian neural network. To match the number of convolutional layers in ALT [15], we use ABA$_{\text{5-layer}}$, *i.e.* ABA with a 5-layer Bayesian convolutional neural network. In the variants ABA$_{\text{5-layer+RandConv}}$ and ABA$_{\text{5-layer+Augmix}}$, the second augmented image is generated by RandConv or Augmix instead of Bayesian neural network.

4.1. Style Shift

We validate our method on two popular style-shift benchmark datasets: (1) **Digits** is composed of digit images from MNIST-10K [26], MNIST-M [12], SVHN [36], USPS [6], SYNTH [11]. Following the setting in [47], MNIST-10K is the source domain containing 10,000 images from MNIST, and the other four datasets are target domains. (2) **PACS** [28] consists of images from four domains: photo, art painting, sketch, and cartoon, and 7 classes. We select one domain as the source domain and the other three as the target domains.

4.1.1 Digits

Baselines. Our baselines include Empirical Risk Minimization (ERM), ADA [47], M-ADA [39], AdvBNN [29], ESDA [46], RandConv [52], Augmix[19] and ALT [15]. For fair comparison with RandConv and ALT, we implement the 1-layer and 5-layer ABA. The classifier architecture for f is DigitNet [47], with $T = 10000$ iterations, batch size 512, learning rate $\gamma = 1e^{-4}$ and Adam optimizer. For ABA g, we set the weight of consistency loss term $\alpha = 3$, adversarial learning rate $\eta = 5e^{-6}$, number of adversarial steps $T_{adv} = 10$, warm-up steps $T_{warmup} = 5$, and factor for KL term of ELBO $\beta = 1$.

Results. Table 1 shows that pixel-level adversarial perturbation methods such as ADA and M-ADA, and the composition of image augmentation method like Augmix, only marginally improve SSDG performance, while AdvBNN even downgrades the performance. However, convolutional-based augmentations, even with just one layer, can significantly enhance performance. Among the 1-layer convolutional augmentations, the weights learned adversarially (ALT) do not perform better than the weights randomly sampled from a fixed distribution (RandConv). However, our 1-layer ABA outperforms both, achieving a 1.69% improvement over RandConv and a 4.34% improvement over 1-layer ALT. Combining a RandConv module does not improve performance much for 1-layer ABA, but

[1] In Tabs. 1 to 3 we highlight the previous best model in gray, variants of ABA better than the previous best in blue, and the best accuracy in **bold**.

Method	MNIST-10K	MNIST-M	SVHN	USPS	SYNTH	Target Avg.
ERM	98.40 (0.84)	58.87 (3.73)	33.41 (5.28)	79.27 (2.70)	42.43 (5.46)	53.50 (4.23)
ADA	N/A	60.41	35.51	77.26	45.32	54.62
M-ADA	99.30	67.94	42.55	78.53	48.95	59.49
ESDA	99.30 (0.10)	81.60 (1.60)	48.90 (5.20)	84.00 (1.20)	62.20 (1.30)	69.12 (2.33)
AdvBNN	98.23 (0.08)	71.79 (0.69)	44.85 (0.55)	46.05 (0.53)	44.99 (0.54)	51.92 (0.51)
Augmix	98.53 (0.18)	53.36 (1.59)	25.96 (0.80)	96.12 (0.72)	42.90 (0.60)	54.59 (0.50)
1-layer convolutional-based augmentations						
RandConv	98.85 (0.04)	**87.76 (0.83)**	**57.62 (2.09)**	83.36 (0.96)	62.88 (0.78)	72.88 (0.58)
ALT$_{\text{1-layer}}$	98.41 (0.15)	72.80 (2.06)	47.07 (1.88)	94.79 (0.88)	66.27 (1.56)	70.23 (1.22)
ALT$_{\text{1-layer+RandConv}}$	98.54 (0.10)	75.77 (1.51)	49.90 (1.62)	95.64 (0.62)	68.61 (1.75)	72.47 (1.18)
ABA$_{\text{1-layer}}$	98.82 (0.09)	78.81 (1.64)	51.88 (1.93)	**96.22 (0.26)**	**71.25 (1.27)**	**74.57 (0.52)**
ABA$_{\text{1-layer+RandConv}}$	98.78 (0.09)	78.62 (0.92)	52.04 (1.13)	96.16 (0.16)	71.23 (0.93)	74.51 (0.70)
3-layer convolutional-based augmentations						
ABA$_{\text{3-layers}}$	98.73 (0.10)	**80.94 (0.39)**	55.88 (0.70)	96.34 (0.54)	73.09 (0.34)	76.56 (0.06)
ABA$_{\text{3-layers+RandConv}}$	98.67 (0.11)	80.05 (0.81)	**56.87 (1.05)**	**96.55 (0.34)**	**73.40 (0.19)**	**76.72 (0.41)**
5-layer convolutional-based augmentations						
ALT$_{\text{5-layer}}$	98.46 (0.27)	74.28 (1.36)	52.25 (1.54)	94.99 (0.68)	68.44 (0.98)	72.49 (0.87)
ALT$_{\text{5-layer+RandConv}}$	98.46 (0.25)	76.90 (1.42)	53.78 (1.97)	95.40 (0.72)	69.40 (1.07)	73.87 (1.03)
ALT$_{\text{5-layer+Augmix}}$	98.55 (0.11)	75.98 (0.89)	55.01 (1.34)	96.17 (0.45)	68.93 (2.17)	74.38 (0.86)
ABA$_{\text{5-layer}}$	98.78 (0.06)	**80.54 (0.53)**	52.45 (1.21)	95.81 (0.47)	70.25 (1.21)	74.76 (0.52)
ABA$_{\text{5-layer+RandConv}}$	98.76 (0.12)	79.69 (0.35)	54.09 (1.27)	**96.42 (0.35)**	**71.55 (0.96)**	75.44 (0.61)
ABA$_{\text{5-layer+Augmix}}$	98.66 (0.16)	80.24 (0.51)	**56.43 (0.59)**	96.14 (0.64)	70.91 (0.83)	**75.93 (0.60)**

Table 1: **SSDG accuracy on Digits dataset.** The source domain is MNIST-10K. The target domains are MNIST-M, SVHN, USPS, SYNTH. We report the mean (and standard deviation) of 5 runs.

still outperforms 1-layer ALT with RandConv by 2.04%. A 5-layer ABA can further improve performance by a small margin 0.19%, similar to the observation of ALT. Adding a RandConv module improves performance by a small margin of 0.68% compared to 5-layer ABA and by 1.57% compared to 5-layer ALT with RandConv. Adding an Augmix module improves by 1.17% compared to 5-layer ABA and by 1.55% compared to 5-layer ALT with Augmix. We achieve state-of-art results by 3-layer ABA with RandConv, with an accuracy of 76.72%. We note that RandConv outperforms all state-of-the-art methods in some domains, such as MNIST-M and SVHN, but does not consistently achieve superior results in other domains.

4.1.2 PACS

Baselines. Our baseline methods include the Empirical Risk Minimization (ERM), JiGEn [3], ADA [47], AdvBNN [29], Augmix [19], Randconv [52], SagNet [35], and ALT [15]. To train the model, we use the pre-trained ResNet18 model and set the training iterations $T = 2000$, with batch size 32, learning rate $\gamma = 4e^{-4}$, SGD optimizer with cosine annealing learning rate scheduler. For AdvBNN, due to the computing complexity, we only bayesianize the linear layers of the networks. For ABA, we set the weight of consistency loss term $\alpha = 3$, adversarial learning rate $\eta = 5e^{-5}$, number of adversarial steps $T_{adv} = 10$, the warm-up steps $T_{warmup} = 4$ and factor for KL term of ELBO $\beta = 0.1$.

Results. Our experiments on the PACS dataset are summarized in Table 2. For PACS, each domain can be considered as a source domain, while the remaining three domains serve as target domains. we report the average accuracy across all target domains for a given source domain in each column, as well as the average accuracy across all four source domains. As PACS contains images with different styles, methods such as SagNet and RandConv that transfer style and preserve shape and texture information can improve performance. In comparison, JiGen and ADA only marginally improve accuracy, while AdvBNN downgrades the performance. Similarly to the Digits dataset, leveraging convolutional-based augmentations provides significant performance improvements, with five variants of ALT performing better than other baseline models. However, our proposed ABA method outperforms ALT, with 1-layer ABA achieving a 2.75% improvement over 1-layer ALT and 5-layer ABA achieving a 2.42% improvement over 5-layer ALT. Adding RandConv modules further improves performance, with ABA$_{\text{1-layer+RandConv}}$ and ABA$_{\text{5-layer+RandConv}}$ achieving improvements of 1.07% and 1.82% over ALT$_{\text{1-layer+RandConv}}$ and ALT$_{\text{5-layer+RandConv}}$, respectively. Adding Augmix modules on 5-layer ABA still performs better than 5-layer ALT with Augmix by 0.83%.

Method	**Photo**	**Cartoon**	**Art**	**Sketch**	**Avg.**
ERM	38.93	70.00	68.83	39.36	54.28
JiGen	41.70	72.23	67.70	36.83	54.61
SagNet	48.53	75.66	73.20	50.06	61.86
ADA	44.63	71.96	72.43	45.73	58.68
AdvBNN	45.93 (0.41)	60.24 (0.95)	75.33 (0.95)	26.19 (1.23)	51.92 (1.15)
Augmix	45.24 (1.12)	74.66 (1.09)	71.47 (0.64)	47.72 (1.72)	60.51 (1.14)
1-layer convolutional-based augmentations					
RandConv	49.80 (4.23)	67.90 (1.55)	69.63 (2.15)	**54.06 (1.96)**	60.34 (2.47)
ALT$_{\text{1-layer}}$	50.83 (2.13)	75.00 (0.62)	73.87 (1.31)	47.83 (1.95)	61.88 (1.50)
ALT$_{\text{1-layer+RandConv}}$	52.24 (0.82)	75.16 (0.67)	73.46 (1.29)	49.21 (2.14)	62.51 (1.23)
ABA$_{\text{1-layer}}$	**54.49 (1.35)**	75.61 (0.89)	75.59 (1.56)	52.84 (2.80)	**64.63 (1.65)**
ABA$_{\text{1-layer+RandConv}}$	52.32 (1.82)	**76.01 (0.56)**	**75.77 (1.64)**	50.20 (1.93)	63.58 (1.49)
3-layer convolutional-based augmentations					
ABA$_{\text{3-layers}}$	**58.86 (0.83)**	**77.49 (0.57)**	75.34 (0.89)	**53.76 (2.46)**	**66.36 (1.19)**
ABA$_{\text{3-layers+RandConv}}$	56.95 (0.80)	77.21 (0.85)	**75.34 (0.52)**	53.52 (0.90)	65.76 (0.15)
5-layer convolutional-based augmentations					
ALT$_{\text{5-layer}}$	54.33 (1.08)	75.96 (1.12)	74.06 (1.09)	50.03 (2.41)	63.60 (1.43)
ALT$_{\text{5-layer+RandConv}}$	55.66 (0.50)	76.23 (0.80)	73.96 (0.54)	50.86 (0.79)	64.18 (0.66)
ALT$_{\text{5-layer+Augmix}}$	55.09 (1.87)	**77.36 (0.73)**	**75.69 (1.21)**	50.72 (1.41)	64.72 (1.30)
ABA$_{\text{5-layer}}$	**59.04 (1.43)**	77.16 (0.35)	74.71 (0.76)	53.18 (2.07)	**66.02 (1.15)**
ABA$_{\text{5-layer+RandConv}}$	57.59 (1.26)	76.66 (0.24)	75.61 (1.02)	**54.12 (1.33)**	66.00 (0.96)
ABA$_{\text{5-layer+Augmix}}$	57.87 (0.22)	77.29 (0.78)	74.70 (0.96)	52.35 (0.03)	65.55 (0.49)

Table 2: **SSDG accuracy on PACS.** Each column is the average accuracy on the target domains trained on the given source domain. We report the mean (and standard deviation) of 5 runs. More details about the accuracy of the source domain to each target domain are in the Appendix.

We achieve the state-of-art results by 3-layer ABA, with an accuracy of 66.36%. We observe that RandConv performs well in the Sketch domain, which is consistent with the intuition that it preserves the shape and texture information.

4.2. Subpopulation Shift

We validate our method on subpopulation shift with the Living17 dataset [41]. Living17 contains images from ImageNet [5] from 17 superclasses, each of which contains 4 subclasses based on WordNet hierarchy [8]. For example, *labrador* and *husky* are subclasses of the superclass *dog*. We choose 2 subclasses in each superclass as the source domain and the rest of the subclasses as the target domain, following the setting in [41]. See the appendix for details.

Baselines. The baselines for Living17 dataset includes the Empirical Risk Minimization (ERM), AdvBNN [29], Augmix [19], Augmax [49], Randconv [52] and ALT [15]. We consider three variants of ALT in our evaluation, by adding RandConv and Augmix module. we do not perform any hyperparameter tuning for Living17 and directly apply identical training settings and hyperparameters from PACS.

Results. Table 3 (left panel) presents our results on the Living17 dataset. We observe that several baseline models (including AdvBNN, Augmix and its adversarial variant Augmax) even worsen the performance on the target domain. RandConv also causes a decrease in performance, although this could be due to it affecting coverage on the source domain, as the source domain accuracy is low. In comparison, both ALT and our proposed method with RandConv module outperform ERM, with ALT achieving a 1.50% improvement and our method achieving a 1.72% improvement, which achieves the best result on Living17 dataset.

4.3. Domain Shift in Medical Imaging

Camelyon17 [1] is the medical imaging dataset for binary classification of tumor detection in the center 32 × 32 region. The dataset is collected from 5 different hospitals. Following the setting in [24], we combine the data from the first 3 hospitals as the source domain and the remaining 2 hospitals as target domains. Note that while some multi-source domain generalization methods utilize domain labels (hospital numbers) for training, we do not use this information, but simply combine data from three hospitals as the single source domain.

Baselines. The baseline models and experiment settings are same as the subpopulation shift experiment, but with difference of using ResNet50 model as the feature extractor.

Results. We present the results of our experiments on the Camelyon17 dataset in Table 3 (right panel). We observe that AdvBNN performs worse than ERM. Augmix,

Method	Source	Target
ERM	95.84 (0.13)	70.97 (0.80)
AdvBNN	92.59 (0.47)	60.76 (0.27)
Augmix	94.58 (0.17)	65.22 (0.90)
Augmax	94.00 (0.30)	63.75 (0.50)
RandConv	87.23 (1.54)	61.18 (1.65)
ALT$_{\text{5-layer}}$	94.98 (0.12)	72.38 (0.84)
ALT$_{\text{5-layer+RandConv}}$	94.91 (0.15)	72.47 (0.63)
ALT$_{\text{5-layer+Augmix}}$	95.03 (0.09)	71.97 (0.62)
ABA$_{\text{5-layer}}$	95.18 (0.31)	72.52 (0.55)
ABA$_{\text{5-layer+RandConv}}$	95.30 (0.24)	**72.69** **(1.16)**
ABA$_{\text{5-layer+Augmix}}$	95.32 (0.12)	72.41 (0.38)

Method	Hospital 1,2,3	Hospital 4	Hospital 5	Target Avg.
ERM	97.96 (0.04)	90.58 (0.63)	82.26 (3.91)	86.42 (1.68)
AdvBNN	96.48 (0.55)	87.30 (2.14)	80.79 (1.69)	84.04 (1.65)
Augmix	96.15 (0.19)	85.92 (1.28)	84.86 (1.58)	85.39 (1.36)
Augmax	95.17 (0.13)	79.61 (1.49)	85.51 (1.69)	82.56 (1.30)
RandConv	97.98 (0.09)	90.64 (1.24)	84.75 (3.94)	87.70 (1.69)
ALT$_{\text{5-layer}}$	96.95 (0.10)	90.82 (0.82)	82.70 (1.22)	86.76 (0.82)
ALT$_{\text{5-layer+RandConv}}$	97.09 (0.01)	91.09 (1.01)	85.80 (1.59)	88.44 (1.16)
ALT$_{\text{5-layer+Augmix}}$	97.06 (0.10)	91.77 (0.42)	86.11 (3.35)	88.94 (1.60)
ABA$_{\text{5-layer}}$	97.29 (0.14)	91.19 (0.64)	85.20 (3.45)	88.19 (1.91)
ABA$_{\text{5-layer+RandConv}}$	97.28 (0.10)	90.89 (0.65)	**88.47** **(0.96)**	89.68 (0.80)
ABA$_{\text{5-layer+Augmix}}$	97.23 (0.06)	**91.85** **(0.78)**	87.92 (0.59)	**89.88** **(0.37)**

Table 3: **SSDG accuracy on Living17 (left panel) and Camelyon17 (right panel).** For Living17, the source domain is the two subclasses in each superclass and the target domain is the remaining two subclasses. For Camelyon17, the source domain is images from hospital 1,2,3 and the target domain is hospital 4 and 5. We report mean (and standard deviation) of 5 runs.

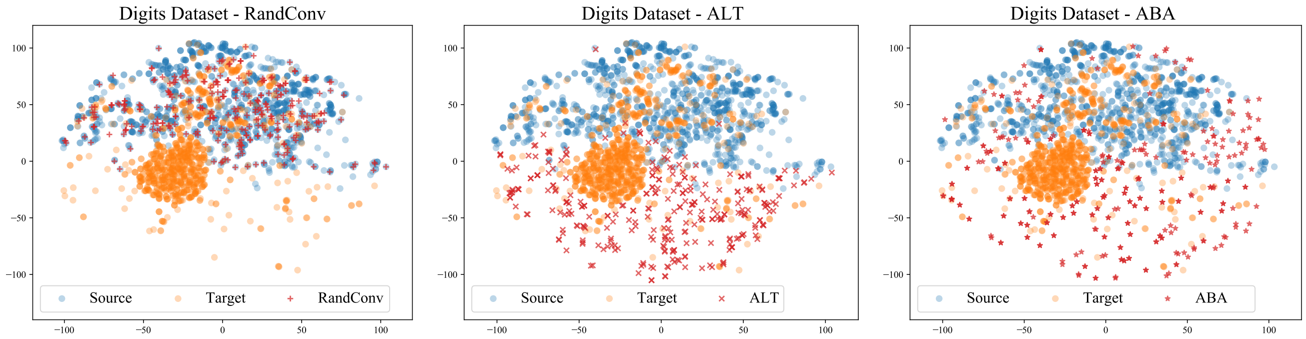

Figure 3: TSNE plot for source domain, target domain and augmented image distribution by RandConv, ALT, ABA.

which composites image augmentation operations, leads to a downgrade in performance, while Augmax, which generates more adversarial samples, performs even worse. However, RandConv can improve the performance, which is consistent with the properties of dataset that rely on shape and texture information. ALT also improves accuracy, while ALT$_{\text{5-layer}}$ performs slightly lower than RandConv about 0.94%, but ALT$_{\text{5-layer+Augmix}}$ achieves an improvement of 1.24% over RandConv. Our ABA$_{\text{5-layer}}$ achieves an improvement of 1.43% compared with ALT$_{\text{5-layer}}$, and similarly, ABA$_{\text{5-layer+RandConv}}$ and ABA$_{\text{5-layer+Augmix}}$ outperforms ALT$_{\text{5-layer+RandConv}}$ and ALT$_{\text{5-layer+Augmix}}$ by 1.24%, 0.94%, achieving the best results on the Camelyon17 dataset.

5. Analysis

In this section we provide further insights on our results, analyses, and ablation studies.

5.1. Key Insights from Results

Limitations of Data Augmentation. First, we observe that every data augmentation method has its limitation on datasets and types of SSDG. For example, AdvBNN, which incorporates the adversarial training of the feature extractor for robustness against the adversarial perturbations, exhibits inferior performance compared to ERM on all four datasets. Augmix performs well on PACS, but not outstanding on the Digits dataset, both of which are style generalization dataset. It even worsens the performance on the generalization of subpopulation and medical imaging. RandConv boosts performance on style generalization, and medical imaging generalization, but hurts the subpopulation shift generalization. Among these baseline models, ALT can keep outperforming most state-of-art methods, but 1) 1 layer ALT does not show significant improvement over RandConv on Digits dataset and 2) ALT sometimes needs additional RandConv or Augmix module to improve the performance.

ABA without additional modules outperforms previous state-of-the-art. The next insight is that even without any explicit module, ABA outperforms all other data augmentation methods, even 1-layer ABA without Randconv module. This indicates that adversarial learned Bayesian convolutional neural networks are powerful for generating augmented images to train a classifier for generalization – we

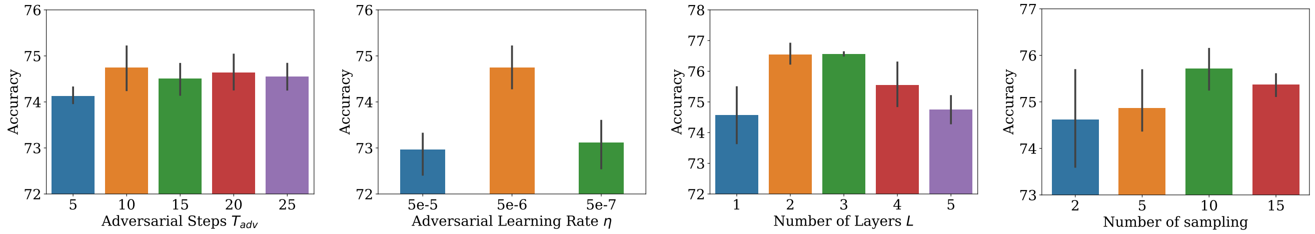

Figure 4: From left to right, we show the impact of hyperparameters on SSDG performance (Digits dataset): adversarial steps, adversarial learning rate, number of layers, and number of sampling per Bayesian convolutional layer.

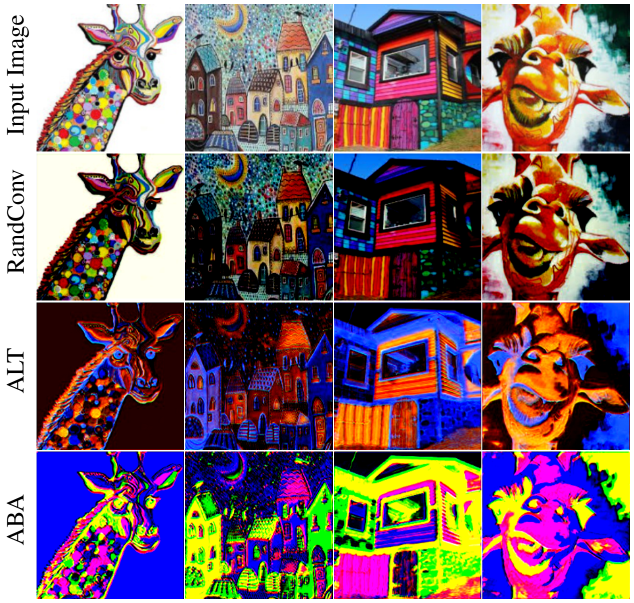

Figure 5: Qualitative comparison of PACS images augmented by RandConv, ALT and our ABA.

can see the benefit of using ABA compared with ALT and RandConv. Moreover, we did not clearly observe the benefits of adding additional RandConv or Augmix module to our ABA method. For example, in the PACS dataset and cartoon as the source domain, adding RandConv can improve performance for the 1-layer ABA, but not for the 5-layer ABA. The addition of Augmix augmentation improves performance on the Digits and Camelyon17 dataset, but does not yield similar improvement on the PACS and Living17 dataset. In order to facilitate a fair comparison with RandConv and ALT which employ 1-layer and 5-layer convolution-based augmentations respectively, we conduct most of our experiments using 1-layer and 5-layer ABA. However, for style shift, our model attains state-of-the-art performance with 3-layer ABA.

ABA results in diverse and distributed augmentations. In Figure 3, we analyze the feature distribution introduced by ABA on Digits dataset and compare it with the source distribution (MNIST-10K), target distribution (SYNTH), and feature distributions from ALT [15] and RandConv [52]. Both ALT and ABA use 1-layer convolutional network. We observe that RandConv can generate some augmented images outside the source domain but still close to it, while ALT, using adversarial learning, mostly generates images outside the source domain. Our ABA method is spread widely across the tSNE [20] space, thanks to the advantage of adversarial learning and Bayesian inference. We also show the qualitative results of augmented images by RandConv, ALT and ABA in Figure 5. We use 1-layer convolutional network for both ALT and ABA. We find that our method can stylize images while still retaining important texture and shape information.

5.2. Ablation study on Hyperparameters

As illustrated in Section 3.1, we control the strength of the adversarial samples by adjusting the adversarial learning rate η and the number of adversarial steps T_{adv}. For ABA, another hyperparameter that determines the model is the number of convolutional layers L. We also explore the impact of the number of sampling per Bayesian convolutional layer on 1-layer ABA. In our paper, to manage computational costs, we adopt a single sampling per layer in the multiple layers BNN. When our method is used without additional augmentation methods, we sample twice per entire BNN network. We investigate the effect of each of these parameters on SSDG in Figure 4. The experiments are conducted on the Digits dataset.

The first plot shows that the number of adversarial steps has little impact on SSDG, once it surpasses 5 steps. While the best results are achieved with 10 adversarial steps, all other results still outperform the previous baseline models. In the second plot, we analyze how the adversarial learning rate affects the results. We find that $\eta = 5e^{-6}$ achieves the best performance, while $\eta = 5e^{-5}$ or $\eta = 5e^{-7}$ may generate adversarial samples that are either too strong or too weak for domain generalization. The third plot demonstrates the importance of the number of ABA layers, with 3-layer ABA achieving the best results. With 1 or 2 layers, the network may not be capable of generating strong enough augmented images, while increasing the number of layers may result in augmented images that are too strong and hurt performance. The last plot shows the impact of the number of sampling of 1-layer ABA. With increasing the number of samplings results in more diverse training samples and improved target accuracy, but the performance starts to decline when the number of sampling exceeds 10. It is clear that these hyperparameters impact the performance of ABA on SSDG.

6. Conclusion

In this paper, we demonstrate how adversarial learning combined with Bayesian convolutional neural network can generate more diverse samples, leading to an improvement in the performance of image classifiers on the single-source domain generalization task. Our method, ABA, outperforms all existing methods on multiple benchmark datasets spanning different types of domain shift. The promising results from this work spark potential future research, such as exploring whether the Bayesian neural network as a feature extractor can improve SSDG.

Acknowledgements

This work was supported by NSF RI grants #1750082 and #2132724. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the funding agencies and employers.

References

[1] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 6

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015. 2

[3] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 2, 5

[4] Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[6] John Denker, W Gardner, Hans Graf, Donnie Henderson, R Howard, W Hubbard, Lawrence D Jackel, Henry Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. In *Advances in Neural Information Processing Systems*, volume 1, 1988. 4

[7] Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40*, pages 187–198. Springer, 2019. 1

[8] Christiane Fellbaum. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier, 2005. 6

[9] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 2

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 2

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 4

[12] 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. 4

[13] 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 *International Conference on Learning Representations*, 2019. 1

[14] Tejas Gokhale, Rushil Anirudh, Bhavya Kailkhura, Jayaraman J Thiagarajan, Chitta Baral, and Yezhou Yang. Attribute-guided adversarial training for robustness to natural perturbations. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 7574–7582, 2021. 1

[15] Tejas Gokhale, Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, Chitta Baral, and Yezhou Yang. Improving diversity with adversarially learned transformations for domain generalization. In *IEEE Winter Conference on Applications of Computer Vision*, pages 434–443, 2023. 2, 3, 4, 5, 6, 8

[16] Tejas Gokhale, Swaroop Mishra, Man Luo, Bhavdeep Singh Sachdeva, and Chitta Baral. Generalized but not robust? comparing the effects of data modification methods on out-of-domain generalization and adversarial robustness. In *Annual Meeting of the Association for Computational Linguistics*, pages 2705–2718. ACL, 2022. 3

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[19] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. 2, 3, 4, 5, 6

[20] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, volume 15, 2002. 8

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. 1

[22] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, volume 28, 2015. 2

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2014. 2

[24] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 2021. 6

[25] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021. 1

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[27] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[28] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, pages 5542–5550, 2017. 4

[29] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-BNN: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations*, 2019. 3, 4, 5, 6

[30] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. The variational fair autoencoder. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2016. 2

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 3

[32] Mazda Moayeri, Kiarash Banihashem, and Soheil Feizi. Explicit tradeoffs between adversarial and natural distributional robustness. In *Advances in Neural Information Processing Systems*, volume 35, pages 38761–38774, 2022. 3

[33] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR, 2017. 2

[34] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013. 1

[35] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021. 2, 5

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems*, 01 2011. 4

[37] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015. 1

[38] Yutian Pang, Sheng Cheng, Jueming Hu, and Yongming Liu. Evaluating the robustness of bayesian neural networks against different types of attacks. *arXiv preprint arXiv:2106.09223*, 2021. 2

[39] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 2, 3, 4

[40] Yunus Saatci and Andrew G Wilson. Bayesian gan. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 2

[41] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. {BREEDS}: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*, 2021. 6

[42] William B Shen, Danfei Xu, Yuke Zhu, Leonidas J Guibas, Li Fei-Fei, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *IEEE International Conference on Computer Vision*, pages 2881–2890, 2019. 2

[43] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019. 3

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2014. 3

[45] Cuong Tran, Keyu Zhu, Ferdinando Fioretto, and Pascal Van Hentenryck. Fairness increases adversarial vulnerability. *arXiv preprint arXiv:2211.11835*, 2022. 3

[46] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation

sets. In *IEEE International Conference on Computer Vision*, pages 7980–7989, 2019. 3, 4

[47] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 1, 2, 3, 4, 5

[48] Haohan Wang, Zexue He, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019. 1

[49] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In *Advances in Neural Information Processing Systems*, volume 34, pages 237–250, 2021. 2, 3, 6

[50] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *International Conference on Learning Representations*, 2018. 2

[51] Zehao Xiao, Jiayi Shen, Xiantong Zhen, Ling Shao, and Cees Snoek. A bit more bayesian: Domain-invariant learning with uncertainty. In *International Conference on Machine Learning*, pages 11351–11361. PMLR, 2021. 1, 2

[52] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2021. 2, 3, 4, 5, 6, 8

[53] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2150, 2015. 1

[54] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 3

[55] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021. 1