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 style shift. ABA outperforms all previous state-of-the-art methods, including pre-specified augmentations, pixel-based and convolutional-based augmentations. Full paper: https://arxiv.org/abs/2307.09520. Code: https://github.com/shengcheng/ABA.

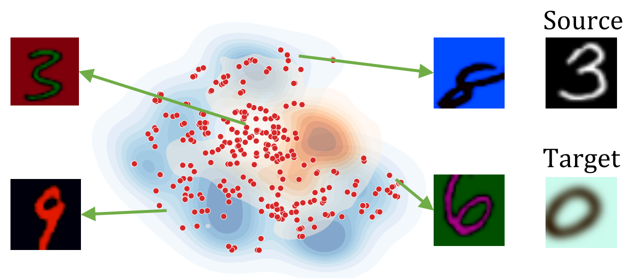

Figure 1: An illustration of the diversity introduced by Adversial Bayesian Augmentations. The blue and orange surfaces represent the source (seen) and target (unseen) domains respectively. The red dots represent the samples augmented by ABA; these augmentations expose the classifier to regions closer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

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 [21, 10, 15]. 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 [20, 17]. 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 [26, 25]. Therefore, there is a need to develop techniques that enable deep neural networks to capture the domain-invariant patterns in the data [15, 23], facilitating improved generalization to out-of-distribution samples.

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 [15, 4, 23]. Prior information about the target domain is also useful to design specific data augmentation methods to tackle domain shift. However, in the single-source domain generalization (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 to simulate the presence of out-of-distribution domains. One way involves learning-free data augmentation methods, such as RandConv [24], Augmix [9] and JiGen [1] – 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 [22], ADA [20], M-ADA [18], and ALT [7]. Although the Bayesian neural networks as the backbone of the classifier show good generalization ability to out-of-distribution samples intrinsically [14, 23, 2], and some papers [19] 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. Specifically, the adversarial learning-based methods, which explore a wider augmentation space, already outperforms learning-free methods [7, 22] 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 ABA's superior performance 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 style generalization. Our method outperforms all existing state-of-art methods.

Algorithm 1: Learning with Adversarial Bayesian Augmentation (ABA)

Input : $\{x_i, y_i\}_{i=1}^N$
Output : Classifier f parameters θ^*

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 θ

2. 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 $\mathcal{T}$.

2.1. Adversarial Bayesian Augmentation

To accomplish SSDG, since no information is available from the target domain $\mathcal{T}$, previous works focus on data augmentation, denoted as g. 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 [24], 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 a sample 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 $\Phi = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$.

The optimization of ABA is formulated as a min-max problem. Initially, we optimize the g network using adversarial optimization 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 used in network initialization [8]. 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 [13].

Starting from a random initialization, the parameters of g are iteratively updated by maximizing the negative of ELBO. In contrast to adv-BNN [13], 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

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)
$\text{ALT}_{\text{1-layer}}$	98.41 (0.15)	72.80 (2.06)	47.07 (1.88)	94.79 (0.88)	66.27 (1.56)	70.23 (1.22)
$\text{ALT}_{\text{1-layer+RandConv}}$	98.54 (0.10)	75.77 (1.51)	49.90 (1.62)	95.64 (0.62)	68.61 (1.75)	72.47 (1.18)
$\text{ABA}_{\text{1-layer}}$	98.82 (0.09)	78.81 (1.64)	51.88 (1.93)	**96.22 (0.26)**	**71.25 (1.27)**	**74.57 (0.52)**
$\text{ABA}_{\text{1-layer+RandConv}}$	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						
$\text{ABA}_{\text{3-layers}}$	98.73 (0.10)	**80.94 (0.39)**	55.88 (0.70)	96.34 (0.54)	73.09 (0.34)	76.56 (0.06)
$\text{ABA}_{\text{3-layers+RandConv}}$	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						
$\text{ALT}_{\text{5-layer}}$	98.46 (0.27)	74.28 (1.36)	52.25 (1.54)	94.99 (0.68)	68.44 (0.98)	72.49 (0.87)
$\text{ALT}_{\text{5-layer+RandConv}}$	98.46 (0.25)	76.90 (1.42)	53.78 (1.97)	95.40 (0.72)	69.40 (1.07)	73.87 (1.03)
$\text{ALT}_{\text{5-layer+Augmix}}$	98.55 (0.11)	75.98 (0.89)	55.01 (1.34)	96.17 (0.45)	68.93 (2.17)	74.38 (0.86)
$\text{ABA}_{\text{5-layer}}$	98.78 (0.06)	**80.54 (0.53)**	52.45 (1.21)	95.81 (0.47)	70.25 (1.21)	74.76 (0.52)
$\text{ABA}_{\text{5-layer+RandConv}}$	98.76 (0.12)	79.69 (0.35)	54.09 (1.27)	**96.42 (0.35)**	**71.55 (0.96)**	75.44 (0.61)
$\text{ABA}_{\text{5-layer+Augmix}}$	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.

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)
$\text{ALT}_{\text{1-layer}}$	50.83 (2.13)	75.00 (0.62)	73.87 (1.31)	47.83 (1.95)	61.88 (1.50)
$\text{ALT}_{\text{1-layer+RandConv}}$	52.24 (0.82)	75.16 (0.67)	73.46 (1.29)	49.21 (2.14)	62.51 (1.23)
$\text{ABA}_{\text{1-layer}}$	**54.49 (1.35)**	75.61 (0.89)	75.59 (1.56)	52.84 (2.80)	**64.63 (1.65)**
$\text{ABA}_{\text{1-layer+RandConv}}$	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					
$\text{ABA}_{\text{3-layers}}$	**58.86 (0.83)**	**77.49 (0.57)**	75.34 (0.89)	**53.76 (2.46)**	**66.36 (1.19)**
$\text{ABA}_{\text{3-layers+RandConv}}$	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					
$\text{ALT}_{\text{5-layer}}$	54.33 (1.08)	75.96 (1.12)	74.06 (1.09)	50.03 (2.41)	63.60 (1.43)
$\text{ALT}_{\text{5-layer+RandConv}}$	55.66 (0.50)	76.23 (0.80)	73.96 (0.54)	50.86 (0.79)	64.18 (0.66)
$\text{ALT}_{\text{5-layer+Augmix}}$	55.09 (1.87)	**77.36 (0.73)**	**75.69 (1.21)**	50.72 (1.41)	64.72 (1.30)
$\text{ABA}_{\text{5-layer}}$	**59.04 (1.43)**	77.16 (0.35)	74.71 (0.76)	53.18 (2.07)	**66.02 (1.15)**
$\text{ABA}_{\text{5-layer+RandConv}}$	57.59 (1.26)	76.66 (0.24)	75.61 (1.02)	**54.12 (1.33)**	66.00 (0.96)
$\text{ABA}_{\text{5-layer+Augmix}}$	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.

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}.

Figure 2: Qualitative comparison of PACS images augmented by RandConv, ALT and our ABA.

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 [24], Augmix [9]. 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 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. The implementation details of each dataset are in Appendix.

3. Experiments

In this section, we validate our method on two popular style-shift benchmark datasets: (1) **Digits** is composed of digit images from MNIST-10K [11], MNIST-M [6], SVHN [16], USPS [3], SYNTH [5]. Following the setting in [20], MNIST-10K is the source domain containing 10,000 images from MNIST, and the other four datasets are target domains. (2) **PACS** [12] 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.

We compare our approach against several state-of-the-art methods [1] using seven variants. For fair comparison with RandConv [24], we use $\text{ABA}_{\text{1-layer}}$, *i.e.* ABA with a 1-layer Bayesian convolutional neural network. To match the number of convolutional layers in ALT [7], we use $\text{ABA}_{\text{5-layer}}$, *i.e.* ABA with a 5-layer Bayesian convolutional neural network. In the variants $\text{ABA}_{\text{5-layer+RandConv}}$ and $\text{ABA}_{\text{5-layer+Augmix}}$, the second augmented image is generated by RandConv or Augmix instead of Bayesian inference.

Results.

For Digits dataset, 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, ALT do not perform better than RandConv. However, our 1-layer ABA outperforms both. A 5-layer ABA performs better than 1-layer ABA and adding a RandConv or Augmix module can further improve performance. We achieve state-of-art results by 3-layer ABA with RandConv to 76.72%.

Our experiments on the PACS dataset are summarized in Table 2. As PACS contains images with different styles, methods such as SagNet and RandConv that 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 four variants of ALT performing better than other baseline models. However, our proposed ABA method outperforms ALT on both 1-layer and 5-layer cases. The addition of RandConv or Augmix modules does not yield further performance improvement, but it still outperforms the corresponding ALT models, respectively. We achieve the state-of-art results by 3-layer ABA, with an accuracy of 66.36%. We show the qualitative results of augmented images by RandConv, ALT and ABA in Figure 2.

4. 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 style 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.

[1] In Tabs. 1 and 2 we highlight the previous best model in gray, variants of ABA better than the previous best in blue, and the best accuracy in **bold**.

References

[1] 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. 1

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

[3] 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

[4] 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

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

[6] 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

[7] 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. 1, 2, 4

[8] 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. 2

[9] 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. 1, 4

[10] 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

[11] 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

[12] 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

[13] 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. 2

[14] 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

[15] 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

[16] 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

[17] 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

[18] 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. 1

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

[20] 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, 4

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

[22] 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. 1, 2

[23] 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

[24] 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. 1, 2, 4

[25] 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

[26] 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