Learning to Augment via Implicit Differentiation for Domain Generalization

Tingwei Wang[1]
tingwei.wang@surrey.ac.uk

Da Li[2]
dali.academic@gmail.com

Kaiyang Zhou[3]
k.zhou.vision@gmail.com

Tao Xiang[1]
t.xiang@surrey.ac.uk

Yi-Zhe Song[1]
y.song@surrey.ac.uk

[1] University of Surrey
Guildford, UK

[2] Samsung AI Center
Cambridge, UK

[3] Nanyang Technological University
Singapore

Abstract

Machine learning models are intrinsically vulnerable to domain shift between training and testing data, resulting in poor performance in novel domains. Domain generalization (DG) aims to overcome the problem by leveraging multiple source domains to learn a domain-generalizable model. In this paper, we propose a novel augmentation-based DG approach, dubbed AugLearn. Different from existing data augmentation methods, our AugLearn views a data augmentation module as hyper-parameters of a classification model and optimizes the module together with the model via meta-learning. Specifically, at each training step, AugLearn (i) divides source domains into a pseudo source and a pseudo target set, and (ii) trains the augmentation module in such a way that the augmented (synthetic) images can make the model generalize well on the pseudo target set. Moreover, to overcome the expensive second-order gradient computation during meta-learning, we formulate an efficient joint training algorithm, for both the augmentation module and the classification model, based on the implicit function theorem. With the flexibility of augmenting data in both time and frequency spaces, AugLearn shows effectiveness on three standard DG benchmarks, PACS, Office-Home and Digits-DG.

1 Introduction

Humans excel at learning visual concepts that are generalizable across different scenarios and environments. For instance, we can easily recognize a dog image no matter whether the image is a realistic photo, a cartoon or even a human drawn sketch. However, though deep neural networks have achieved great success in many computer vision tasks, their ability to generalize to novel data distributions remains rather limited. This hampers the wide deployment of deep learning models in real-world applications. The root of poor generalization is the domain shift problem [8, 32, 58, 41] which is known by machine learning

researchers for decades. One solution to the domain shift problem is unsupervised domain adaptation (UDA) [1, 13, 16, 28], which exploits unlabeled target domain data for domain adaptation. Although UDA avoids target domain data annotation, it still needs access to the target domain data to perform model adaptation for each target domain. Domain generalization (DG) [5, 8, 18, 19, 51, 54] is motivated to solve this drawback of UDA. Given multiple source domains, the goal of DG is to learn a model that can generalize well to any unseen target domain without any model adaptation.

Most existing DG methods are either feature alignment or meta-learning based. Feature alignment based methods borrow ideas from the domain adaptation community to align features across source domains [11, 15, 20, 25, 53, 55]. These methods focus on minimizing the divergence between the source domains in feature space, which ensures that the extracted representations are domain agnostic. In contrast, meta-learning methods expose models to domain shift during training [2, 9, 24, 27]. The underlying idea is to split training data into meta-train and meta-test sets without overlapping domains. A DG model is trained on the meta-train set in a way such that its loss on the meta-test set is also low.

Recently, data augmentation based DG methods have attracted increasing attention [43, 47, 49, 51, 52, 55]. Existing methods design image synthesizers to synthesize images that do not belong to any existing source domains. They are designed with the assumption that diversified source domains enable the model to learn more generalizable features. However, existing augmentation DG methods normally require some complicated design of learning objectives, such as differing the augmented images from the source data, maintaining the fidelity of the augmented images, retaining the same semantic meaning of the augmented image to the vanilla image. More importantly, there is no guarantee that the augmented images, when used for training the main classification model, can ensure the model generalizes well to an unseen domain.

In this paper, we propose a novel meta learning based augmentation method for DG. Different from the existing augmentation based DG methods, we treat the image augmentation module as the hyperparameters of a classification model and optimize both jointly. Crucially, the augmentation module is optimized explicitly to help the model generalize to a novel domain. To guarantee generalization, we expose the model to simulated domain shift and meta learn the augmentation module to minimize the generalization error of the model. However, optimizing the hyperparameters of the model is non-trivial when these hyperparameters are actually parameters of a deep CNN (i.e., the augmentation module) and thus in the order of millions. In particular, the typical bilevel hyperparameter optimization will produce second-order gradients which poses a serious computational challenge [29, 30, 36]. To address this computational issue, we employ the implicit function theorem (IFT) [30] to avoid storing the inner loop update trajectories which is prohibitively expensive. Specifically, we create episodes from minibatch images of source domains during training. In each episode, we randomly split the images from different source domains into pseudo source and pseudo target domains. The classification model is optimized by minimizing the classification loss on pseudo source domain data. The hyperparameters (augmentation module) are then optimized using the Neumann series approximated IFT by minimizing a validation loss on the pseudo target data. Frequency-based data augmentation has recently shown promising performance in domain generalization [46]. Flexibly, our AugLearn is not only applicable to augment the input image in time space, but also in frequency space. We can simply feed the frequency spectrum, such as obtained by discrete cosine transformation (DCT), of the input image into the AugLearn and then inverse the augmented frequency spectrum to the time space – this variant is dubbed AugLearn-F. Our proposed augmentation module is model-agnostic and

can be applied to any base DG methods.

Our **contributions** are summarized as follows: (1) We propose a novel DG framework in which the augmentation module is viewed as hyperparameters of the model and optimized efficiently using the implicit function theorem (IFT). (2) Different from the most conventional data augmentation methods, focusing on the augmentation in the time space only, our AugLearn is capable of augmenting the input data in both the time and frequency spaces. (3) Our proposed module is model-agnostic and applicable to any base DG methods. Extensive experiments are carried out, and the results show that our method achieves the state of the art performance on two popular DG benchmarks, PACS and Digits-DG.

2 Related Work

Domain generalization. The DG problem was first introduced in [5] in which they proposed a kernel-based DG approach. Since then DG has been receiving increasing attention [44, 54] from the research community due to its importance to practical machine learning applications. DG models aim at extracting general representations that can perform well on unseen target domains. In general, the existing DG methods can be categorised into three groups, namely domain alignment, data augmentation and meta-learning.

(1) The alignment based methods are mainly inspired by the domain adaptation literature. The main idea is to learn representations that are domain-agnostic among source domains such that the representations can generalize to any unseen target domain. Muandet *et al.* [34] developed Domain-Invariant Component Analysis (DICA), which adopts a kernel-based optimization algorithm that reduces the discrepancy across source domains, to learn domain invariant features. Li *et al.* [22] proposed a low-rank parameterized CNN for learning domain agnostic features. Li *et al.* [25] extended adversarial autoencoders by imposing the Maximum Mean Discrepancy (MMD) measure to align distributions among different source domains. Motiian *et al.* [33] employed a Siamese architecture to map different source domains to a discriminative embedding subspace, where the mapped features are semantically aligned and maximally separated.

(2) The existing data augmentation based DG methods mainly focused on two folds, the image level and feature level augmentations. Yue *et al.* [47] proposed a domain randomization based augmentation to diversify the input images and forced the model to learn domain invariant features. Zhou *et al.* [53] developed a learnable data perturbation module to generate novel images to improve the model generalization, and a novel domain generator using optimal transport as a followup [52]. Recently researchers found that feature augmentation is effective in improving model generalization [26, 55]. MixStyle [55] assumed the feature statistics represent the domain style information and proposed to interpolate feature representations following Mixup [50]. Li et al. [26] lately found that a simple feature perturbation using Gaussian noise worked pretty well on improving domain generalization. All these augmentation based methods require explicit objectives, such as generating new domain images while retaining the same semantic meaning. Furthermore, those methods focus on diversifying the source domains and this is not promising to improve the generalization ability of the models. Our proposed method optimizes the augmentation module implicitly to be general to unseen domain.

(3) Meta-learning is widely applied on improving few shot learning [12, 57] and has now been exploited to tackle domain shift problems [2, 9, 23]. [23] proposed to mimic the domain shift during training by splitting the source domains into meta train and meta test such that the

training model can learn to learn how to address DG. Balaji *et al.* [2] then reformulated the DG problem as meta-learning a generalizable regularizer. Li *et al.* [24] developed an episodic training paradigm to improve DG by manipulating domain specific feature extractors and classifiers. Dou *et al.* [9] designed a meta learning algorithm to improve DG by maintaining the inter-class knowledge consistency across source domains. Du *et al.* [10] proposed to meta learn a normalization to tackle the statistic changes among source and target domains.

Implicit function theorem. Bilevel optimization is typical in hyperparameter optimization [4] and meta learning [12, 36]. However, it normally triggers the second order derivatives during the back propagation. Implicit function theorem (IFT) has been explored to address such computational issue [29, 30, 36]. However, the conventional IFT requires the computation of inverse hessian which is intractable in deep neural networks. Therefore, various approximations have been proposed to mitigate this problem. [30] proposed to approximate the hessian with the identity matrix, while [36] uses Conjugate Gradient (CG) to approximate the vector-inverse hessian product. More recently, [29] proposed a novel approximation using Neumann series, which is used in our paper.

3 Methodology

Problem setup. In the DG setting, it is normally assumed there are multiple source domains $\mathcal{D} = \{D_1, \ldots, D_S\}$, where each D_i typically consists of many data pairs $\{x, y\}_i^N$. Then a model F_θ, i.e. CNN, is trained on the source domains, and then tested on an unseen target domain D_{S+1}.

ERM baseline. The most straightforward DG method is to train a model using the empirical risk minimization on the source domain data. The formulation is as follows

$$\arg\min_\theta \frac{1}{|\mathcal{D}|} \sum_{x,y \sim \mathcal{D}} \ell_{ce}(F_\theta(x), y), \tag{1}$$

where ℓ_{ce} is the cross entropy loss. After training, F_θ will be deployed for model inference.

Naive data augmentation. Data augmentation as a regularization has shown its effectiveness on improving the model generalization [21]. Typically the augmentation operation is sampled and constructed stochastically from different hand crafted operations, such as `flipping`, `rotation`, `jittering` and `coloring`, denoted as $\mathcal{A}(\cdot)$. Then the augmented images will be used to train the model, thus Eq. 1 becomes

$$\arg\min_\theta \frac{1}{|\mathcal{D}|} \sum_{x,y \sim \mathcal{D}, A \sim \mathcal{A}} \ell_{ce}(F_\theta(A(x)), y). \tag{2}$$

3.1 Data Augmentation Module as Hyperparameters

The typical data augmentation is made up of different hand crafted operations, which have demonstrated their effectiveness in standard supervised learning but may not be optimal to domain generalization tasks. Therefore, we treat the data augmentation module as hyperparameters of our classification model and optimize it against domain shift during training. The overall scheme is illustrated in Figure 1.
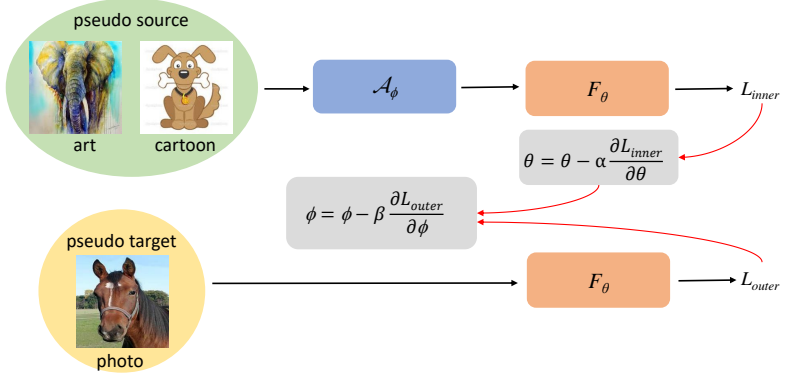
Figure 1: The overview of our proposed AugLearn method. We treat the augmentation module $\mathcal{A}_\phi$ as hyperparameters of the classification model F_θ. The classification model is updated on pseudo source domains, and the augmentation module is optimized on the pseudo target domain under the condition that θ^* is optimal to pseudo source domains. The red arrows in this figure denote the gradient flow through the second-order differentiation.

Augmentation module. We exploit a simple UNet [39] as our augmentation module, which is demonstrated in Figure 2. Specifically, this UNet consists of three convolutional blocks, a transpose convolution layer and a convolution layer. The convolutional block is composed of two convolution layers, each followed by a ReLU activation. In addition, a max-pooling layer is attached to the first convolutional block. The UNet is formulated as $\mathcal{A}_\phi(\cdot)$ parameterized by ϕ.
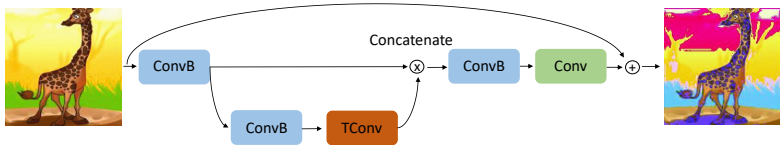

Figure 2: The illustration of the augmentation module. ConvB, TConv and Conv in this figure are convolutional block, transpose convolution layer and convolution layer, respectively.

Bilevel optimization. In order to optimize our augmentation module, we create episodes using our source domain data during training. we split the source domain data $\mathcal{D}$ into pseudo source $\mathcal{D}^{\hat{p}src}$ and pseudo target $\mathcal{D}^{\hat{p}trg}$ at each mini-batch step. Then the bilevel optimization is conducted, including an inner loop optimization on $\mathcal{D}^{\hat{p}src}$ which is formulated as

$$\theta^* = \arg\min_\theta \frac{1}{|\mathcal{D}^{\hat{p}src}|} \sum_{x,y \sim \mathcal{D}^{\hat{p}src}} \ell_{ce}(F_\theta(\mathcal{A}_\phi(x)), y), \tag{3}$$

and an outer loop optimization on $\mathcal{D}^{\hat{p}trg}$

$$\phi = \arg\min_\phi \frac{1}{|\mathcal{D}^{\hat{p}trg}|} \sum_{x,y \sim \mathcal{D}^{\hat{p}trg}} \ell_{ce}(F_{\theta^*}(\phi(x)), y). \tag{4}$$

Let us break down the computation in Eq.4. If the inner and outer loop losses are denoted as L_{inner} and L_{outer}, then the hypergradients of L_{outer} w.r.t. the augmentation module $\mathcal{A}_\phi$ is computed as

$$
\begin{aligned}
\frac{\partial L_{outer}(\phi)}{\partial \phi} &= \frac{\partial L_{outer}}{\partial \phi} + \frac{\partial L_{outer}}{\partial \theta} \frac{\partial \theta^*}{\partial \phi} \\
&= \underbrace{\frac{\partial L_{outer}}{\partial \phi}}_{\text{direct grad.}} + \underbrace{\frac{\partial L_{outer}(\phi, \theta^*(\phi))}{\partial \theta^*(\phi)} \times \frac{\partial \theta^*(\phi)}{\partial \phi}}_{\text{indirect grad.}}.
\end{aligned}
\tag{5}
$$

In our case, the **direct grad** is zero, thus we only need to calculate the **indirect grad**. However, the indirect grad is hard to compute as the best-response Jacobian term $\frac{\partial \theta^*(\phi)}{\partial \phi}$ needs the computation through the conditional training trajectories of $\min \theta^*(\phi)$ given ϕ [29]. One can use IFT to approximate the best-response Jacobian as follows

$$
\frac{\partial \theta^*(\phi)}{\partial \phi} = -\left[\frac{\partial^2 L_{inner}}{\partial \theta \partial \theta^T}\right]^{-1} \times \frac{\partial^2 L_{inner}}{\partial \theta \partial \phi^T}.
\tag{6}
$$

Neumann series based IFT. Inverting a Hessian matrix in high dimension is intractable. In our algorithm, we employ the Neumann series to approximate the Hessian inversion as per [29]. The formulation is as follows

$$
\left[\frac{\partial^2 L_{inner}}{\partial \theta \partial \theta^T}\right]^{-1} = \lim_{i \to \infty} \sum_{j=0}^{i} \left[I - \frac{\partial^2 L_{inner}}{\partial \theta \partial \theta^T}\right].
\tag{7}
$$

Thanks to the efficiency brought by this approximation, the augmentation module $\mathcal{A}_\phi$ and the model F_θ can be optimized efficiently during training. The pipeline is summarized as Algorithm 1.

Algorithm 1: Learning to Augment for DG

Input: Domain $\mathcal{D}$
Init: Classification model parameters θ. Augmentation module parameters ϕ.
  Hyperparameters α, β
for *ite **in** iterations* **do**
  　Split: $\mathcal{D}^{\hat{p}src}, \mathcal{D}^{\hat{p}trg} \leftarrow \mathcal{D}$
  　for *ite **in** inner iterations* **do**
  　　Inner-loop: Compute gradients $\frac{\partial L_{inner}}{\partial \theta}$
  　　Inner-loop optimization: Update parameters $\theta = \theta - \alpha \frac{\partial L_{inner}}{\partial \theta}$
  　end
  　Outer-loop: Compute gradients $\frac{\partial L_{outer}(\phi)}{\partial \phi}$ according to Eq. 5, Eq. 6 and Eq. 7 in
  　　the main body.
  　Outer-loop optimization: Updated parameters $\phi = \phi - \beta \frac{\partial L_{outer}}{\partial \phi}$
end
Output: θ

Augmentation in frequency space. Our AugLearn is also capable of augmenting the input data in frequency space. To this end, we first convert the image into the frequency space using DCT, and then apply the augmentation module to the frequency spectrum. The inner loop optimization equation 3 is then reformulated as

$$\theta^* = \arg\min_{\theta} \frac{1}{|\hat{\mathcal{D}^{src}}|} \sum_{x,y \sim \hat{\mathcal{D}^{src}}} \ell_{ce}(F_{\theta}(\mathcal{T}_{inv}(\mathcal{A}_{\phi}(\mathcal{T}(x)))),y), \tag{8}$$

where $\mathcal{T}(\cdot)$ is DCT, and $\mathcal{T}_{inv}(\cdot)$ is inverse DCT. This variant is named as AugLearn-F. The overall training and inference of AugLearn-F are the same as the vanilla AugLearn method.

3.2 Inference

At model inference, the augmentation module is disabled. Given an input from the unseen domain D_{S+1}, the prediction is computed as

$$\hat{y} = F_{\theta}(x), x \sim D_{S+1}. \tag{9}$$

As $F_{\theta}(x)$ is trained with an augmentation module, which is optimized to generate augmented images against domain shift, the precision of $\hat{y}$ from our model is thus guaranteed.

4 Experiments

4.1 Experimental Setup

Datasets and settings. We evaluate our approach on three commonly used DG benchmark datasets, namely PACS [22], Office-Home [42] and Digits-DG [53]. (1) PACS is composed of four domains, which are art, cartoon, photo and sketch, with 9,991 images in total. There are seven classes for each one of these domains, and the domain shift mainly comes from dramatic style changes. (2) Office-Home consists of four domains including artistic, clipart, product and real-world. There are 65 object classes and approximately 15,500 images in total. Images from different domains differ in viewpoint, background and image style. (3) Digits-DG includes MNIST, MNIST-M, SVHN and SYN, which differ in font style, stroke and background. There are ten classes in each domain, and each class has 600 images. We report the top-1 classification accuracy averaged over five runs with different random seeds. Due to the space constraint, the Digits-DG results are available in supplementary material.

Baselines. We compare our method with the current state of the art DG methods including CCSA [33], MMD-AAE [25], CrossGrad [40], JiGen [6], DDAIG [53], L2A-OT [52] and MixStyle [55]. CCSA and MMD-AAE align the features from different source domains to a unified space and train a model based on those aligned features. CrossGrad is based on the domain classification guided image augmentation. JiGen introduces jigsaw solving to DG as an auxiliary task. DDAIG and L2A-OT are designed to generate novel domain images to improve the model generalization ability. MixStyle diversifies inputs by mixing the styles in feature space. We also compare with ERM, which serves as a strong baseline for DG.

4.2 Evaluation on PACS

Implementation. Following [6], we use the ImageNet-pretrained ResNet18 as the feature extractor with a followed softmax classifier. All the images are resized to 224×224. The networks are trained with SGD, with initial learning rate of $1e-3$, batch size of 16 and weight decay of $5e-4$ for 50 epochs. The learning rate is decayed by 0.1 at the 30th epoch and the 20th epoch for classification optimizer and augment-optimizer, respectively.

Results. The experimental results on PACS is shown in Table 1 (left). First, we observe that our proposed augmentation modules AugLearn and AugLearn-F outperform the ERM baseline with remarkable accuracy margins 4.6% and 4.8% respectively, demonstrating the efficacy of our proposed algorithms. More interestingly, our AugLearn variants are complementary with a feature augmentation based DG method, MixStyle. Incorporating our AugLearn(-F), it improves 1.3% (1.2%) accuracy over the vanilla MixStyle. Our proposed augmentation algorithm differs from DDAIG and L2A-OT regarding optimization objectives and training strategy. The results show that optimizing the augmentation module explicitly with the simulated domain shift can be much more effective than optimizing it using the complicated learning objectives on the source domain data. Our AugLearn(-F) outperforms all the other methods on cartoon and sketch domains, which are two most challenging held out domains, especially with a noticeable improvement (5.6%) on sketch. These results explain our assumption, that a pure image augmentation may be beneficial to model performance but is not robust to large domain shifts.

Method	PACS					Office-Home				
	Art	Cartoon	Photo	Sketch	Average	Artistic	Clipart	Product	Real World	Average
ERM	78.5	75.2	96.2	67.9	79.5	58.4	49.2	74.1	76.3	64.5
CCSA	80.5	76.9	93.6	66.8	79.4	59.9	49.9	74.1	75.7	64.9
MMD-AAE	75.2	72.7	96.0	64.2	77.0	56.5	47.3	72.1	74.8	62.7
CrossGrad	79.8	76.8	96.0	70.2	80.7	58.4	49.4	73.9	75.8	64.4
JiGen	79.4	75.3	96.0	71.6	80.5	53.0	47.5	71.5	72.8	61.2
DDAIG	**84.2**	77.0	95.3	83.1	74.7	59.2	52.3	74.6	76.0	65.5
L2A-OT	83.3	78.2	**96.2**	82.8	73.6	**60.6**	50.1	**74.8**	**77.0**	65.6
MixStyle	84.1	78.8	96.1	75.9	83.7	58.7	53.4	74.2	75.9	65.5
ERM+AugLearn	82.9	78.8	94.5	80.1	84.1 (+4.6)	58.9	53.3	74.3	76.0	65.6 (+1.1)
ERM+AugLearn-F	81.9	**79.2**	95.3	80.7	84.3 (+4.8)	58.5	**54.2**	73.2	75.1	65.3 (+0.8)
MixStyle+AugLearn	84.1	79.0	95.2	**81.5**	**85.0** (+1.3)	59.3	53.5	74.6	76.0	**66.0** (+0.5)
MixStyle+AugLearn-F	83.9	**79.2**	95.4	81.0	84.9 (+1.2)	59.8	52.7	**74.8**	75.6	65.7 (+0.2)

Table 1: Leave-one-domain-out generalization results on PACS and Office-Home.

4.3 Evaluation on Office-Home

Implementation. We use the training, validation splits following [55]. We train the model on the training set from the source domains and test the trained model on the held out test domain. The training details are the same as those of PACS dataset.

Results. As shown in Table 1 (right), our proposed method achieves again the state of the art performance on this benchmark, further demonstrating the effectiveness of our proposed AugLearn. Specifically, our AugLearn(-F) improves over the ERM baseline with a 1.1% (0.8%) accuracy margin. Meanwhile, our method achieves comparable results with

L2A-OT, the recent augmentation based DG method. Again on the most challenging held out domain clipart, we achieve the best performance over all other competitors, demonstrating the robustness of our AugLearn(-F) against large domain shift. Our AugLearn(-F) still complements with MixStyle enabling a 0.5% (0.2%) accuracy gain over vanilla MixStyle.

5 Further Analysis

5.1 Ablation Study

We conduct further experiments to analyze our proposed AugLearn(-F). We attribute the performance of our AugLearn to the simulated domain shift and meta learning. We thus compare it with a simple variant (w/o meta learning), which updates the classification model and the augmentation module on the pseudo source and pseudo target domains respectively, without bilevel optimization. From the results in Table 2, we observe that optimizing the augmentation module during training is indeed helpful with 3.2% and 3.5% accuracy improvements over the ERM baseline with regard to AugLearn and AugLearn-F, respectively. Adding meta learning with domain shift improves the model performance further by 1.4% and 1.3% accuracy margins. These results demonstrate the efficacy of our proposed meta learning pipeline.

5.2 Visualization of Augmented Images

Figure 3 illustrates the outputs of the augmentation module trained on two used DG benchmarks. We can see that the generated images (middle) by AugLearn are different from the original images (left) dramatically. The augmentation module not only changes the background but also diversifies the appearance of objects. These illustrative results show that our AugLearn indeed generates augmented images which are different from the vanilla ones but with the same semantic meaning without using explicit learning objectives. We attribute this to our meta learning pipeline, which enables easy training. Another interesting observation is that the images generated by AugLearn-F do not change much from the original images, i.e. Figure 3 left v.s. right, while AugLearn-F still can improve the model performance clearly. This shows it is more effective to augment images in frequency space than that in time space.

5.3 Side Benefit of Adversarial Defence

We also include the investigation of our models against adversarial attacks, such as FGSM [17]. After training the DG models, i.e. ERM, ERM with strong augmentations (Cutout [7], CutMix [48] and DropBlock [14]) and AugLearn variants, we attack the trained DG model using FGSM attack with different strengths. From the results in Figure 4, we can see that after incorporating our AugLearn modules the attack success rate decreases significantly though the models are trained on clean data only. AugLearn(-F) enables the model to be more robust against FGSM attack while Cutout and Cutmix fail. Interestingly, we can see that injecting the AugLearn-F module during the training brings more model robustness than AugLearn against FGSM attack. It is found that the adversarial attacks conduct more perturbations in the middle and high frequencies of the vanilla image [45]. Therefore, augmenting the input images in the frequency space during training may give more robustness to the trained model against potential adversarial attacks than augmenting in the time space.
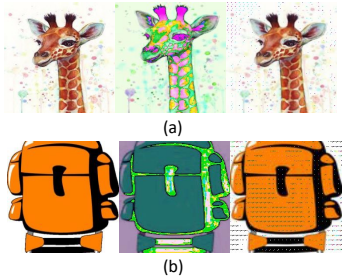
Method	Art	Cartoon	Photo	Sketch	Avg.
ERM	78.5	75.2	96.2	67.9	79.5
AugLearn	82.9	78.8	94.5	80.1	84.1
- w/o ML	81.6	76.3	93.8	79.0	82.7
AugLearn-F	81.9	79.2	95.3	80.7	84.3
- w/o ML	81.2	76.5	94.2	79.9	83.0

Table 2: Ablation study results on PACS

Figure 3: Synthesized examples on PACS (a) and Office-Home (b). Left: raw, middle: AugLearn generated, right: AugLearn-F generated.

Figure 4: Success rate of FGSM attack on PACS.

6 Compact Augmentation Module

We also analyse the extra parameters introduced in the model training of different augmentation based DG methods. From the numbers in Table 3, we can see that our AugLearn introduces extremely few trainable parameters compared with CrossGrad, DDAIG and L2A-OT. Specifically, we can see that CrossGrad doubles the total training parameters by introducing the augmentation module. Then, the method with the second most extra parameters is L2A-OT, which adds 3.70m parameters. The tier three is DDAIG with

Method	backbone	extra parameters
CrossGrad		11.17m
DDAIG	11.18m	0.23m
L2A-OT		3.70m
AugLearn		6.65k

Table 3: Number of trainable parameters of different methods.

0.23m extra parameters. Nevertheless, they brought many orders of magnitude more parameters compared to our AugLearn. We assume this is due to the nature of our proposed meta learning pipeline, such that our augmentation module is effective though trained with much fewer parameters.

7 Conclusion

We have presented a novel data augmentation based DG method, termed AugLearn. AugLearn treats the augmentation module as the model hyperparameters and optimizes it with meta learning. Our AugLearn is light-weight, model-agnostic and applicable to any base DG methods (verified with two different DG methods). More inherently, our AugLearn module is capable of augmenting data in both the time and frequency spaces. Extensive experiments demonstrate that AugLearn variants achieve the state of the art performance on two popular DG benchmarks. Qualitative visualizations further explain that our AugLearn is able to generate augmented images which are different from the vanilla images in terms of both the foreground and background, with a simple meta learning objective.

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013.

[2] Y. Balaji, S. Sankaranarayanan, and R. Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008, 2018.

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W.n Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[4] Y. Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12 (8):1889–1900, 2000.

[5] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. volume 24, pages 2178–2186, 2011.

[6] F. M. Carlucci, A. D'Innocente, S. Bucci, and T. Caputo, B.and Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[7] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[8] Z. Ding and Y. Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing*, 27(1):304–313, 2017.

[9] Q. Dou, D. C. Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6450–6461, 2019.

[10] Y. Du, X. Zhen, L. Shao, and C. G. Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*, 2020.

[11] S. Erfani, M. Baktashmotlagh, M. Moshtaghi, X. Nguyen, C. Leckie, J. Bailey, and R. Kotagiri. Robust domain generalisation by enforcing distribution invariance. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 1455–1461. AAAI Press, 2016.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[14] G. Ghiasi, T. Lin, and Q. V. Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018.

[15] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. 2015.

[18] Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020.

[19] S. Jeon, K. Hong, P. Lee, J. Lee, and H. Byun. Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 22–31, 2021.

[20] X. Jin, C. Lan, W. Zeng, and Z. Chen. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*, 2020.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. volume 25, pages 1097–1105, 2012.

[22] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[23] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[24] D. Li, J. Zhang, Y. Yang, C. Liu, Y. Song, and T. M. Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019.

[25] H. Li, J. S. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[26] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[27] Y. Li, Y. Yang, W. Zhou, and T. Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019.

[28] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1417, 2014.

[29] J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

[30] J. Luketina, M. Berglund, K. Greff, and T. Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*, pages 2952–2960. PMLR, 2016.

[31] T. Matsuura and T. Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11749–11756, 2020.

[32] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[33] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017.

[34] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

[35] Se. Otálora, M. Atzori, V. Andrearczyk, A. Khan, and H. Müller. Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology. *Frontiers in bioengineering and biotechnology*, 7:198, 2019.

[36] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 2019.

[37] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[38] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[40] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi. Generalizing across domains via cross-gradient training. In *Proceedings of the ICLR*, 2018.

[41] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in neural information processing systems*, 2012.

[42] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[43] R. Volpi and V. Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7980–7989, 2019.

[44] J. Wang, C. Lan, C. Liu, Y. Ouyang, W. Zeng, and T. Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021.

[45] Z. Wang, Y. Yang, A. Shrivastava, V. Rawal, and Z. Ding. Towards frequency-based explanation for robust CNN. *CoRR*, abs/2005.03141, 2020. URL https://arxiv.org/abs/2005.03141.

[46] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[47] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019.

[48] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[49] S. Zakharov, W. Kehl, and S. Ilic. Deceptionnet: Network-driven domain randomization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 532–541, 2019.

[50] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[51] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. J. Wood, H. Roth, A. Myronenko, D. Xu, et al. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7):2531–2540, 2020.

[52] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020.

[53] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020.

[54] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[55] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. Domain generalization with mixstyle. In *Proceedings of the ICLR*, 2021.