Randomized Adversarial Style Perturbations for Domain Generalization

Taehoon Kim[1] Bohyung Han[1,2]
ECE[1] & IPAI[2], Seoul National University
{kthone, bhhan}@snu.ac.kr

Abstract

We propose a novel domain generalization technique, referred to as Randomized Adversarial Style Perturbation (RASP), which is motivated by the observation that the characteristics of each domain are captured by the feature statistics corresponding to its style. The proposed algorithm perturbs the style of a feature in an adversarial direction towards a randomly selected class. By incorporating the perturbed styles into training, we prevent the model from being misled by the unexpected styles observed in unseen target domains. While RASP is effective for handling domain shifts, its naïve integration into the training procedure is prone to degrade the capability of learning knowledge from source domains due to the feature distortions caused by style perturbation. This challenge is alleviated by Normalized Feature Mixup (NFM) during training, which facilitates learning the original features while achieving robustness to perturbed representations. We evaluate the proposed algorithm via extensive experiments on various benchmarks and show that our approach improves domain generalization performance, especially in large-scale benchmarks.

1. Introduction

One of the major drawbacks of machine learning models compared to human intelligence is the lack of adaptivity to distribution shifts. While humans easily make correct decisions even on unseen domains, deep neural networks often exhibit significant performance degradation on the data from unseen domains. The lack of robustness to novel domains restricts the applicability of neural networks to real-world problems since it is implausible to build a training dataset that covers all possible domains and follows the true data distribution. Therefore, learning domain-invariant representations with limited data in the source domains is critical for deploying deep neural networks in practical systems.

Domain Generalization (DG) attempts to train a machine learning model that is robust to unseen target domains, using data from source domains. The most straightforward way to achieve this goal is to expose the model to various domains during the training procedure. To stretch the coverage of the source domains, recent approaches often employ data generation strategies [16, 19, 22, 26, 29, 32–35]. While they have shown promising results on generalization ability, many of them require additional information about data such as domain labels for individual instances [22, 33–35] or even extra network components such as generators and domain classifiers [22, 29, 33, 34]. However, the additional information including domain labels is unavailable in general and the need for architectural support increases computational complexity and training burden. There exist a few approaches that do not require extra information about data or additional network modules [16, 19], but they are limited to straightforward feature augmentations by stochastically adding trivial noise. While [22, 26] adopt adversarial data augmentation techniques, they impose perturbations in image space without style disentanglement, which incurs higher computational complexity and inferior generalization capabilities.

This paper presents a simple yet effective data augmentation technique based on adversarial attacks in the feature space for domain generalization. The proposed approach does not require architectural modifications or domain labels but relies on feature statistics in the intermediate layers. Our work is motivated by the observation that each visual domain differs in its feature statistics given by instance normalization, which corresponds to the style of a feature. Based on this observation, existing works attempt to learn style-agnostic networks robust to domain shifts via style augmentations [12, 19, 35]. Although they do not require additional networks [19, 35], they are limited to using simple augmentation techniques with no feedback loop in the augmentation process, leading to suboptimal performance. While StyleNeophile [12] augments novel styles that have different distributions from the source domain using the information observed in the previous iterations, its simple style diversification objective is weak for improving performance on unseen styles in the target domain.

Although our approach follows the same assumption as [12, 19, 35], its objective for style augmentation is unique

in the sense that it actively synthesizes hard examples to improve trained models. To this end, inspired by adversarial attacks [7, 17, 27], the proposed method, referred to as Randomized Adversarial Style Perturbations (RASP), adversarially augments the styles of features so that the corresponding examples deceive the network to be misclassified. Unlike the other methods based on adversarial attacks toward the fixed target label [22, 32], we draw random labels for attacks to ensure the plausibilty of the augmented styles. While the features with modified styles strengthen the generalization ability on unseen domains, they might neglect crucial information observable in the source domains since the style augmentation is prone to disturb the representations of perturbation-free examples. To compensate for this, we propose Normalized Feature Mixup (NFM) technique based on mixup [30]. Instead of applying the naïve feature mixup technique, NFM combines the normalized representations of perturbed and perturbation-free features. By integrating normalized features given by NFM, we successfully maintain the representations from the source domains while taking advantage of style augmentation based on RASP.

Our contributions are summarized as follows:

- We present a unique style augmentation technique, referred to as RASP, for domain generalization based on adversarial learning. This method is free from any architectural modifications or the need for the domain label of each example.
- We introduce a novel feature mixup method, NFM, which allows us to maintain knowledge from source domains while facilitating the adaptation to fresh data via robust domain augmentation.
- The proposed approach consistently demonstrates outstanding generalization ability in multiple standard benchmarks, especially in large-scale datasets.

The rest of this paper is organized as follows. We first review previous works about domain generalization in Section 2, and our main algorithm based on RASP and NFM is discussed in Section 3. We present experimental results from the standard benchmarks in Section 4 and conclude this paper in Section 5.

2. Related Works

In the pursuit of robust Domain Generalization (DG), a central challenge is to develop models that generalize to unseen target domains using only source domains for training. Existing approaches address the DG problem by using the following techniques: 1) meta-learning 2) data augmentation, 3) feature statistics manipulation, and 4) flat minima seeking. This section summarizes the technical details of each of the four categories.

Meta-learning approaches The algorithms in this line of research formulate the domain generalization task as a meta-learning problem [2, 6, 15] by splitting the source domains into the meta-train and meta-test sets. Using these sets, they adopt the learn-to-learn schemes of meta-learning for the generalization on unseen domains. However, they rely heavily on the assumption that the diversity of the source domains is large enough to effectively cover unseen domains, which may not hold in real-world scenarios.

Data augmentation approaches The methods in this category, which deal with domain shifts by introducing new images belonging to new domains for training [22, 23, 26, 28, 29, 32–34], and can be divided into two groups. One group uses generative models to increase the number of training examples [29, 33, 34], which induces extra computational complexity and instability of training. The approaches in the other direction rely on image perturbation for data augmentation [22, 26, 28, 32]. However, some of them [22, 28] still require auxiliary neural networks for the regularization. While some image perturbation approaches [22, 26, 32] employ adversarial augmentation techniques similar to our method, they operate on image space without style disentanglement, leading to high computational complexity and inferior performance compared to the proposed method.

Feature statistics manipulation approaches The methods in this type take advantage of the observation that the feature statistics capture characteristics of the visual domains [12, 19, 21, 35]. DSON [21] introduces domain specific normalization layer using the weighted sum of batch and instance normalization statistics. MixStyle [35] generates novel domain features by mixing feature statistics from different images while pAdaIN [19] randomly permutes the statistics of each feature before every batch normalization layer. Similarly, SFA [16] adopts a feature-level augmentation technique that applies simple stochastic linear transforms. Unlike [19, 35], StyleNeophile [12] augments styles that have different distributions from those in previous iterations. Although StyleNeophile [12] does not rely on simple stochasticity, there is no guarantee that the augmented styles will be useful for the generalization in target tasks.

Flat minima seeking approach It is well-known that flat minima in the objective function facilitates learning robust models to the variations of input data. SWAD [3] exploits this property and proposes the stochastic weight averaging strategy [11] tailored for domain generalization; the technique finds flat minima for enhancing the generalization ability to domain shift.

3. Proposed Approach

This section describes the technical details of the proposed Randomized Adversarial Style Perturbations (RASP)

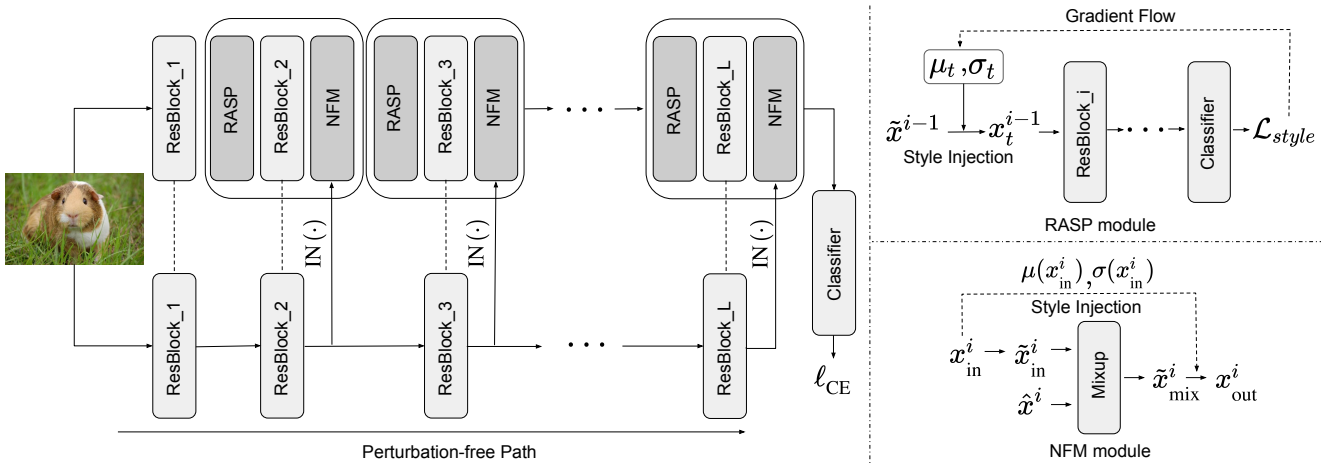

Figure 1. Overall framework of our algorithm, Randomized Adversarial Style Perturbations (RASP) with Normalized Feature Mixup (NFM). Our model runs two parallel paths, one with RASP+NFM and the other without the modules. Before each block in the RASP+NFM path, we apply the RASP module to augment novel styles. RASP adjusts the style of each feature by minimizing the loss with respect to a random target class different from the ground-truth. After passing through the RASP module, followed by a ResBlock, NFM is performed to regularize the style-augmented features. The perturbation-free path injects the style normalized information into the NFM module. Note that the common network components in the two paths share their parameters.

and the Normalized Feature Mixup (NFM) methods.

3.1. Background

Instance normalization and style Recent studies on neural style transfer [10, 24] discover that the style of a feature can be captured by the instance-specific channel-wise mean and standard deviation. Instance Normalization (IN) [24] removes the effect of styles on features by normalizing features as follows:

$$\text{IN}(z) = \gamma \frac{z - \mu(z)}{\sigma(z)} + \beta, \tag{1}$$

where z is the feature for an input example x, (γ, β) are learnable affine parameters, and $(\mu(z), \sigma(z))$ are instance-specific channel-wise statistics computed by

$$\mu(z) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} z_{h,w}, \tag{2}$$

$$\sigma(z) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (z_{h,w} - \mu(z))^2}. \tag{3}$$

AdaIN [10] allows the style of an input feature z' to be transferred to the content of another input z by replacing the affine parameters of IN with the feature statistics of the style image z' as follows:

$$\text{AdaIN}(z, z') = \sigma(z') \frac{z - \mu(z)}{\sigma(z)} + \mu(z'). \tag{4}$$

Gradient-based attacks Adversarial attacks [7, 13, 17, 18, 27, 31] trick a neural network by injecting imperceptible small noise to an example. A popular way to generate such perturbations is to utilize the gradient information of the network. Given an image-label pair (x, y) and model parameters θ, FGSM [7] computes the optimal perturbation of a linearized cost function, which is given by

$$x = x + \epsilon \, \text{sign}(\nabla_x J_\theta(x, y)), \tag{5}$$

where ϵ is the magnitude of perturbation and $J_\theta(\cdot, \cdot)$ is a task-specific loss function. I-FGSM [13] extends FGSM [7] by using multiple iterations as

$$x_0 = x, \tag{6}$$

$$x_{t+1} = \text{clip}_{x,\epsilon} \left[x_t + \alpha \, \text{sign}(\nabla_x J_\theta(x_t, y)) \right], \tag{7}$$

where α is the step size for each iteration and $\text{clip}_{x,\epsilon}[\cdot]$ denotes a clipping operation that restricts the magnitude of a perturbation from the original image only up to ϵ. We adopt an unclipped and targeted version of I-FGSM as our baseline of adversarial attacks.

3.2. Overall Framework

Let (x, y) denote an image and class label pair sampled from an arbitrary source domain. We adopt Empirical Risk Minimization (ERM) using the standard cross entropy loss, ℓ_{CE}, as our baseline. Our goal is to train a feature extractor f_θ and a classifier g_ϕ parameterized by θ and ϕ, respectively, which are robust to domain shifts. To apply our methods, we divide f_θ into L residual blocks as $f_\theta :=$

$f_\theta^L \circ f_\theta^{L-1} \circ \cdots \circ f_\theta^1$. When we train the proposed network, we consider an additional forwarding path—a perturbation-free path—with shared weights in parallel and make the two paths interact with each other.

To introduce the challenging styles during training, we incorporate the Randomized Adversarial Style Perturbations (RASP) module, which will be discussed in Section 3.3, before each block of the backbone network with a probability of 0.5. When RASP is applied, we also perform Normalized Feature Mixup (NFM) after each block with a probability of 0.5. It prevent the augmented features from being deviated too much from the original features. Figure 1 illustrates the overall framework of the proposed approach.

3.3. Randomized Adversarial Style Perturbations (RASP)

The proposed method augments the styles of features in each block of a network, employs the style-augmented examples for adversarial training, and improves the robustness of the trained model to the features regardless of their styles. While the existing style perturbation methods [12, 19, 35] are helpful for learning style-agnostic feature extractors, we argue that simply generating diverse examples is suboptimal for training models and that the target direction in style augmentation is critical for training domain-agnostic models using a limited number of data with new styles.

RASP generates examples that meet two desirable properties—style difficulty and plausibility. First, the augmentation process is conducted in a way that provides challenging styles, to prevent the model trained with the styles from being deceived by the unexpected distribution shifts. Second, since the proposed algorithm perturbs an example towards one of the target classes other than its ground-truth, the generated examples tend to be more realistic than those obtained by simply reducing the score corresponding to the ground-truth label. In addition to the target selection strategy, we adopt a threshold to terminate the RASP iterations when the prediction score for the ground-truth label falls below the threshold; it ensures that the augmented styles are within the desired range aligned with realistic images.

The training procedure with RASP is simple. Given a feature z^{i-1} for an input x after the $(i-1)^{\text{st}}$ block and a randomly sampled target label y_{target} different from the original label y, we compute the style loss $\mathcal{L}_{\text{style}}$ at each attack iteration t as

$$\mathcal{L}_{\text{style}} = \ell_{\text{CE}}\left(g_\phi \circ f_\theta^L \circ \cdots \circ f_\theta^i(z_t^{i-1}), y_{\text{target}}\right), \quad (8)$$

where $z_t^{i-1} = \tilde{z}^{i-1} \cdot \sigma_t + \mu_t$ is the denormalized vector and ℓ_{CE} is standard crossentropy loss. Notice that $\tilde{z}^{i-1}$ is the instance normalized feature derived from z^{i-1} and (μ_t, σ_t) characterizes the style at the t^{th} iteration, where $(\mu_0, \sigma_0) = (\mu(z^{i-1}), \sigma(z^{i-1}))$. Given a step size ϵ, the style of a feature is updated in a way that decreases $\mathcal{L}_{\text{style}}$ as follows:

$$\mu_{t+1} = \mu_t - \epsilon \cdot \|\mu_0\|_2 \cdot \text{sign}(\nabla_{\mu_t} \mathcal{L}_{\text{style}}) \quad (9)$$

$$\sigma_{t+1} = \sigma_t - \epsilon \cdot \|\sigma_0\|_2 \cdot \text{sign}(\nabla_{\sigma_t} \mathcal{L}_{\text{style}}) \quad (10)$$

as long as the following condition is met:

$$\text{softmax}(g_\phi \circ f_\theta^L \circ \cdots \circ f_\theta^i(z_t^{i-1}))_y \geq \tau. \quad (11)$$

where τ is the score threshold of the ground-truth class for terminating the RASP iterations. Note that the step sizes in RASP for updating the mean and the standard deviation are proportional to their magnitudes as in (9) and (10).

3.4. Normalized Feature Mixup (NFM)

Although RASP provides plenty of effective novel styles that strengthen generalization performance on unseen domains, it degrades the capability of learning features from source domains since perturbed styles may deviate excessively from the original ones. To compensate for this phenomenon, we propose the Normalized Feature Mixup technique (NFM) technique, which ensembles instance-normalized features from both the perturbation-free path and the RASP path. By doing this, NFM preserves the knowledge from the source domain while learning robust representations by taking advantage of style augmentations.

Instead of using the features before the instance normalization, which may change the augmented styles back to the original one by mixup, NFM performs mixup with the normalized features (content features) and then applies augmented styles to the mixed normalized features via denormalization. Formally, given a feature from the i^{th} block z_{in}^i, we obtain an instance-normalized feature, $\tilde{z}_{\text{in}}^i$, and its augmented style obtained from the RASP path. NFM module mixes $\tilde{z}_{\text{in}}^i$ with an instance-normalized feature from the perturbation-free path in the i^{th} block, $\hat{z}^i$, to get a mixed normalized feature, $\tilde{z}_{\text{mix}}^i$, and denormalizes it with the augmented style $(\mu(z_{\text{in}}^i), \sigma(z_{\text{in}}^i))$ to obtain the final output, z_{out}^i, as follows:

$$\tilde{z}_{\text{mix}}^i = \alpha \cdot \hat{z}^i + (1-\alpha) \cdot \tilde{z}_{\text{in}}^i, \quad (12)$$

$$z_{\text{out}}^i = \tilde{z}_{\text{mix}}^i \cdot \sigma(z_{\text{in}}^i) + \mu(z_{\text{in}}^i), \quad (13)$$

where $\alpha \sim \text{Beta}(0.1, 0.1)$ determines the mixup ratio.

3.5. Inference

While our method utilizes an additional forwarding path and optimization steps during training, we only use the perturbation-free path for inference. Therefore, it does not incur additional computational cost at the inference time.

4. Experiments

We demonstrate the performance of the proposed algorithm on standard benchmarks of domain generalization and

Table 1. Performance on DomainNet with two different backbone networks. RASP+NFM presents outstanding accuracy in this large-scale dataset. The bold-faced numbers indicate the best performance.

(a) Results of ResNet18 on DomainNet

Method	Additional components	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ERM	—	56.6	18.4	45.3	12.5	57.9	38.8	38.3
MetaReg [2]	Domain label, Task network	53.7	21.1	45.3	10.6	**58.5**	42.3	38.6
DMG [4]	Domain label, Mask predictor	60.1	18.8	44.5	14.2	54.7	41.7	39.0
StyleNeophile [12]	—	60.1	17.8	46.5	14.6	55.4	45.3	40.0
RASP+NFM (ours)	—	**60.4** ± **0.2**	**22.6**± **0.1**	**50.2**± **0.2**	**17.2**± **0.3**	56.8± 0.3	**48.5**± **0.4**	**42.6**

(b) Results of ResNet50 on DomainNet

Method	Additional components	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ERM	—	64.0	23.6	51.0	13.1	64.5	47.8	44.0
MetaReg [2]	Domain label, Task network	59.8	25.6	50.2	11.5	65.5	50.1	43.6
DMG [4]	Domain label, Mask predictor	65.2	22.2	50.0	15.7	59.6	49.0	43.6
StyleNeophile [12]	—	66.1	21.4	51.4	15.3	61.7	51.8	44.6
SWAD [3]	—	66.0	22.4	53.5	16.1	**65.8**	55.5	46.5
RASP+NFM (ours)	—	**66.5**± **0.2**	**27.4**± **0.2**	**55.2**± **0.2**	**16.9**± **0.3**	63.7± 0.1	**53.8**± **0.3**	**47.2**

analyze the characteristics of our approach in comparison with existing techniques.

4.1. Datasets and Evaluation Protocol

We evaluate the proposed algorithm on DomainNet [20], Office-Home [25], and PACS [14], which are standard benchmarks for domain generalization. We set DomainNet, which is a large-scale dataset in terms of the number of classes and examples, as our primary target benchmark since verification in a large-scale benchmark is essential to confirm whether the proposed algorithm is applicable in real-world problems. DomainNet contains 586,475 images of 6 domains (Clipart, Infograph, Painting, Quickdraw, Real, and Sketch) and 345 classes. Office-Home contains 15,558 images of 65 classes from 4 different domains (Artistic, Clipart, Product, and Real world) while PACS, the smallest dataset, consists of 9,991 images of 4 domains (Photo, Art paint, Cartoon, and Sketches) and 7 classes (dog, elephant, giraffe, guitar, horse, house and person).

We use the source domain validation set as the validation set for model selection, following the *"training-domain validation"* criterion of DomainBed [8]. All the results are average classification accuracy over five runs with different random seeds.

4.2. Implementation Details

We employ ResNet18 or ResNet50 [9] pretrained on ImageNet [5] as our backbone network architectures. We use the SGD optimizer with a learning rate of 0.0005 decayed by 0.1 after 30 epochs. The batch size is set to 32 and the number of epochs for training is set to 60.

For our approach, we set the threshold of the ground-truth class probability τ to 0.8, the step size ϵ to $\frac{2}{255}$, and the number of attack iterations to 5 for all datasets. Note that ϵ is rescaled by multiplying $\frac{64}{\text{channel size}}$ to maintain the size of perturbations at each layer. RASP and NFM are applied to the 2nd, 3rd, and 4th residual blocks, before and after each block, respectively, as illustrated in Figure 1. All of the experiments are performed on a single TITAN-XP GPU using VESSL [1].

4.3. Results on DomainNet

We present the quantitative results of RASP with NFM on DomainNet in comparison with other existing methods. DomainNet is the most challenging dataset for DG since it is the largest and there are substantial domain shifts between domains, compared to other datasets. Despite these challenges, the proposed algorithm consistently achieves significant performance gains in all cases except the Real domain with the ResNet18 and ResNet50 backbones as shown in Table 1. Note that improvements in the domains with severe distribution shifts (Infograph, Painting, Quickdraw, Sketch) are more salient than the mild domains (Clipart, Real). This result implies that the optimization for the worst-case styles is particularly helpful in the case that there is a large domain gap between source and target domains.

4.4. Results on Office-Home

To validate the effectiveness of our approach, we conducted experiments on an additional dataset, Office-Home. Table 2 clearly shows that RASP with NFM outperforms other methods on this dataset. The proposed method exhibits greater performance improvement in more challenging target domains; this is consistent with the observation

Table 2. Performance on Office-Home with two different backbone networks. Note that the shaded rows indicate the algorithms that use different hyperparameters for individual target domains, resulting in overestimated accuracies. The bold-faced numbers indicate the best performance among the methods under fair comparisons without domain-specific hyperparameter turning.

(a) Results of ResNet18 on Office-Home

Method	Additional components	Art	Clipart	Product	Real	Avg.
ERM	—	59.0	48.4	72.5	75.5	63.9
CrossGrad [22]	Domain classifier/label	58.4	49.4	73.9	75.8	64.4
MixStyle [35]	Domain label	58.7	53.4	74.2	75.9	65.5
StyleNeophile [12]	—	59.6	55.0	73.6	75.5	65.9
RASP+NFM (ours)	—	**59.7 ± 0.4**	**57.6± 0.9**	**75.2± 0.3**	**76.7± 0.4**	**67.3**
DDAIG [33]	Generator	59.2	52.3	74.6	76.0	65.5
ADVTSRL [29]	Generator	60.7	52.9	75.8	77.2	66.7

(b) Results of ResNet50 on Office-Home

Method	Additional components	Art	Clipart	Product	Real	Avg.
ERM	—	64.7	58.8	77.9	79.0	70.1
CrossGrad [22]	Domain classifier/label	67.7	57.7	79.1	80.4	71.2
MixStyle [35]	Domain label	64.9	58.8	78.3	78.7	70.2
SWAD [3]	—	66.1	57.7	78.4	80.1	70.6
RASP+NFM (ours)	—	**68.8± 0.4**	**61.7± 1.0**	**79.8± 0.2**	**80.9± 0.3**	**72.8**
DDAIG [33]	Generator	65.2	59.2	77.7	76.7	69.7
ADVTSRL [29]	Generator	69.3	60.1	81.5	82.1	73.3

Table 3. Performance on PACS with the ResNet18 backbone network. Note that the shaded rows of the table indicate the algorithms that use different hyperparameters for individual target domains. The bold-faced numbers indicate the best performance among the methods that do not require the hyperparameter tuning specific to target domains.

Results of ResNet18 on PACS

Method	Additional components	Art	Cartoon	Photo	Sketch	Avg.
ERM	—	75.1	74.2	95.6	68.4	78.3
CrossGrad [22]	Domain classifier/label	79.8.	76.8	96.0	70.2	80.7
MixStyle [35]	Domain label	84.1	78.8	**96.1**	75.9	83.7
StyleNeophile [12]	—	84.4	78.4	94.9	**83.3**	**85.5**
RASP+NFM (ours)	—	**84.6± 0.5**	**79.8± 0.5**	94.1± 0.4	80.1± 1.1	84.7
DDAIG [33]	Generator	84.2	78.1	95.3	74.7	83.1
ADVTSRL [29]	Generator	85.8	80.7	97.3	77.3	85.3

in DomainNet that RASP is useful especially when there exists a larger domain discrepancy between the source and the target. ADVTSRL [29] and DDAIG [33] are optimized with a separate set of hyperparameters for each target domain and the comparisons with these methods are unfair.

4.5. Results on PACS

We also evaluate RASP with NFM on the PACS dataset and present the results in Table 3. The proposed method is competitive with all the compared methods even without additional components such as domain classifiers or labels. As mentioned earlier, ADVTSRL [29] and DDAIG [33] use domain-specific hyperparameters, which makes the comparisons with our method unfair.

4.6. Ablation studies

As pointed out in DomainBed [8], selecting proper hyperparameters is a major issue for domain generalization since the target domain data is inaccessible during training. To ensure the applicability of the proposed algorithm to real-world domain generalization scenarios, we analyze the effect of each hyperparameter on target domain accuracies and source domain validation accuracies. Also, we test the various options of the proposed algorithm to further validate the effectiveness of each component.

Number of attack iterations We study how the performance of RASP is influenced by the number of attack iterations, T. Table 4(a) presents the evaluation results on

Table 4. Ablation study results on the variations of attack iterations (T) and attack termination threshold (τ) with ResNet18 on Office-Home.

Ablation types	Variations	Art	Clipart	Product	Real	Avg.	Source Acc.
(a) Attack iteration (T)	1	59.5	53.0	75.6	76.9	66.3	83.7
	2	**60.8**	55.0	**76.0**	77.2	67.3	84.1
	3	60.5	56.5	75.5	**77.4**	**67.5**	**84.3**
	4	60.2	56.9	75.6	77.0	67.4	83.8
	5	59.7	**57.6**	75.2	76.7	67.3	83.5
(b) Threshold (τ)	0.0	57.4	58.0	72.4	74.8	65.7	82.4
	0.2	58.0	**58.6**	73.4	75.2	65.7	83.2
	0.4	58.6	58.5	74.2	76.0	66.3	83.5
	0.6	59.2	58.0	74.7	76.5	67.1	**83.8**
	0.8	**59.7**	57.6	**75.2**	**76.7**	**67.3**	83.5

Table 5. Ablation study results on the step size (ϵ) with ResNet18 on Office-Home.

Step size	Art	Clipart	Product	Real	Avg.	Source Acc.
$\epsilon = 0.5/255$	60.1	53.2	**76.1**	76.9	66.6	83.9
$\epsilon = 1/255$	**61.0**	55.4	75.9	**77.2**	**67.4**	**84.2**
$\epsilon = 2/255$	59.7	57.6	75.2	76.7	67.3	83.5
$\epsilon = 3/255$	59.0	**58.2**	74.0	76.0	66.8	83.3

Table 6. Ablation study results on the location where RASP is applied with ResNet18 on PACS.

RGB	Res2	Res3	Res4	Art	Cartoon	Photo	Sketch	Avg.
				75.1	74.2	95.6	68.4	78.3
ASA [32]				75.8	76.3	95.7	67.4	78.8
CrossGrad [22]				79.8	76.8	**96.0**	70.2	80.7
✓				79.3	76.7	**96.0**	73.5	81.4
	✓			82.2	77.9	95.1	76.9	83.0
		✓		82.1	77.3	94.7	77.7	83.0
			✓	80.1	78.0	95.5	68.2	80.5
	✓	✓		83.2	78.0	94.2	**80.2**	83.9
	✓		✓	84.5	79.2	94.7	76.4	83.7
		✓	✓	83.4	79.5	94.4	77.6	83.7
	✓	✓	✓	**84.6**	**79.8**	94.1	80.1	**84.7**

Office-Home by varying the number of attack iterations. Since the augmented examples with more adversarial iterations have more challenging styles, the classification accuracy on relatively easy target domains, *e.g.,* Product and Real, decreases as the number of iterations increases while the accuracy on more difficult target domains, *e.g.*, Clipart, benefits from more iterations. An important observation from Table 4(a) is that target domain accuracies have positive correlations with source domain validation accuracies; one can easily select the proper T by observing source domain validation accuracies.

Stopping criterion of attack Our algorithm stops the adversarial attack if the score of the ground-truth drops below a threshold, regardless of the number of attack iterations. The threshold is a hyperparameter that balances the difficulty and plausibility of the synthesized style. Table 4(b) presents the accuracies of our model by varying the threshold values. Overall, our models with large thresholds achieve better results in general compared to the models with small ones. However, a small threshold is rather effective for the domains with large domain shifts, *e.g.*, Clipart, because the augmented examples provide challenging styles. Note that, if we set the threshold as too low, the overall performance is degraded substantially. Similar to the ablation study about the number of attack iterations, the accuracies in the target domain and the source domain validation set have positive correlations when we vary the threshold τ. Hence, it is reasonable to select the proper hyperparameters based on the validation accuracy in the source domain.

Step size Table 5 illustrates the results on Office-Home under the change of step size ϵ. As the ϵ increases, augmented styles get diverse while the plausibility is reduced. Consequently, challenging target domains, *e.g.*, Clipart, exhibit improved accuracies, while easy domains, *e.g.*, Product and Real, and source domains witness degraded performances. Note that there is a positive correlation between the accuracies of source domain validation sets and target domains.

Attack locations within models We evaluate the proposed algorithms by applying our style augmentation technique to multiple different layers in the network. Table 6 demonstrates that the performance of the proposed method varies greatly depending on the location of style augmentation. Our algorithm provides a novel perspective on domain generalization in the sense that it attacks the feature statistics (styles) of the intermediate features instead of input images. Table 6 clear shows inferior performance of ASA [32] and RGB-level style attacks in our method,

Table 7. Ablation study results on other variations of our algorithm with ResNet18 on PACS.

Ablation types	Variations	Art	Cartoon	Photo	Sketch	Avg.
(a) Augmentation Objective	Ours	**84.6**	**79.8**	**94.1**	80.1	**84.7**
	RASP_{GT}	84.3	78.1	93.3	**80.6**	84.1
(b) NFM	w/o NFM	82.6	79.3	92.9	**80.1**	83.7
	Style Mixup	83.4	79.0	93.5	79.5	83.9
	Mixup	84.1	78.7	93.7	79.1	83.9
	Ours	**84.6**	**79.8**	**94.1**	**80.1**	**84.7**

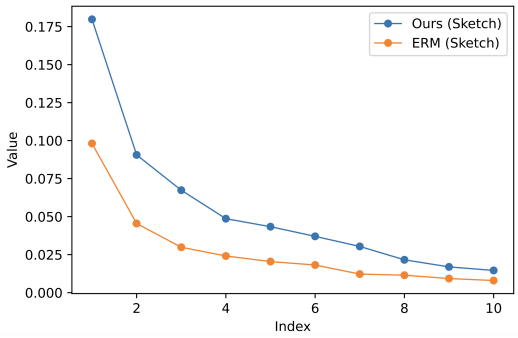

Figure 2. Eigenvalues of the covariance matrix of the channel mean vectors after the 2nd ResBlock for the test set of the target domain, sketch of the PACS dataset.

highlighting the importance of feature-level augmentation. Also, the comparison between RGB-level style attacks and CrossGrad [22] shows the importance of style attacks with style disentanglement. Note that CrossGrad [22] does not disentangle styles but requires domain labels and domain classifiers.

Attack objectives RASP randomly selects a target class for augmentation. To show the effectiveness of this strategy, we test another option for augmentation directions. One simple way is to add perturbation in the direction of decreasing the prediction score of the ground-truth, which is referred to as RASP_{GT}. As shown in Table 7(a), such a direction is clearly outperformed by the proposed scheme. We argue that this occurs because simply decreasing the score of the ground-truth label leads to a degenerate example, indicating an unrealistic image caused by untargeted style perturbations. RASP_{GT} produces more challenging but unrealistic styles. Consequently, RASP_{GT} improves accuracy on the Sketch domain, which is the most challenging domain while degrading performance on other domains. This modified objective of the adversarial attack degrades the overall performance.

Variations of NFM Table 7(b) presents the results from the various options related to the NFM modules. When mixup is applied to styles or features instead of normalized features, classification accuracy significantly drops. This is partly because the variations of our mixup strategy fail to benefit from the novel style generation capability of RASP. When NFM is not employed, the accuracy of easy target domains, *e.g.*, art and photo, is significantly degraded. These results support our arguments that NFM maintains the knowledge learned from the source domains.

Effects on diversity of styles We plot the eigenvalues of the covariance matrix of channel mean vectors after the 2nd ResBlock using the test set of the target domain, the sketch domain in PACS in this case. Figure 2 illustrates that the proposed method with consistently larger eigenvalues allows the network to observe more diverse styles than the model based on ERM.

5. Conclusion

We presented a simple yet effective style augmentation framework for domain generalization, called RASP, based on adversarial attacks. RASP augments the styles that deceive the network by attacking the model itself. Training models using the examples with the augmented styles is helpful for improving the generalization ability on unseen domains by making the feature extractors robust to the style changes. In addition to this idea, we also proposed NFM to make the perturbed features have the desired properties and further enhance the domain generalization performance. RASP with NFM does not require any architectural modifications or domain labels and can be easily incorporated into the existing baselines. Extensive experiments show that the proposed algorithm consistently improves the generalization performance on unseen target domains across multiple datasets.

Acknowledgements This work was supported by Samsung Electronics Co., Ltd. [IO210917-08957-01], the National Research Foundation of Korea (NRF) grant [No. 2022R1A2C3012210] and Institute of Information & communications Technology Planning & Evaluation (IITP) grant [No.2022-0-00959, (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making; No.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], funded by the Korea government (MSIT).

References

[1] Jaeman An. Model development with vessl, 2023. Software available from vessl.ai. 5

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018. 2, 5

[3] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *NeurIPS*, 2021. 2, 5, 6

[4] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*, 2020. 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *ECCV*, 2020. 2

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015. 2, 3

[8] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. 5, 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[11] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. 2

[12] Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *CVPR*, 2022. 1, 2, 4, 5, 6

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 3

[14] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 5

[15] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2

[16] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *ICCV*, 2021. 1, 2

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 2, 3

[18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 3

[19] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: reducing the bias towards global statistics in image classification. In *CVPR*, 2021. 1, 2, 4

[20] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 5

[21] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, 2020. 2

[22] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018. 1, 2, 6, 7, 8

[23] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 2017. 2

[24] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[25] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 5

[26] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*. 1, 2

[27] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 2, 3

[28] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, 2021. 2

[29] Fu-En Yang, Yuan-Chia Cheng, Zu-Yun Shiau, and Yu-Chiang Frank Wang. Adversarial teacher-student representation learning for domain generalization. *NeurIPS*, 2021. 1, 2, 6

[30] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2

[31] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *ICLR*, 2018. 3

[32] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. In *NeurIPS*, 2022. 1, 2, 7

[33] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020. 1, 2, 6

[34] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 1, 2

[35] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 1, 2, 4, 6