RANDOMIZED ADVERSARIAL STYLE PERTURBATIONS FOR DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

While deep neural networks have shown remarkable progress in various computer vision tasks, they often suffer from weak generalization ability on unseen domains. To tackle performance degradation under such domain shifts, Domain Generalization (DG) aims to learn domain invariant features applicable to unseen target domains based only on the data in source domains. This paper presents a simple yet effective approach for domain generalization via style perturbation using adversarial attacks. Motivated by the observation that the characteristics of each domain are captured by the feature statistics corresponding to style, we propose a novel domain generalization technique, referred to as Randomized Adversarial Style Perturbations (RASP). The proposed algorithm augments the styles of features to deceive the network outputs towards randomly selected labels during training and prevents the network from being misled by the unexpected styles observed in unseen target domains. While RASP is effective to handle domain shifts, its naïve integration into the training procedure might degrade the capability of learning knowledge from source domains because it has no restriction on the perturbations of representations. This challenge is alleviated by Normalized Feature Mixup (NFM), which facilitates learning the original features while achieving robustness to perturbed representations via their mixup during training. We evaluate the proposed algorithm via extensive experiments on various benchmarks and show that our approach improves domain generalization ability, especially in large-scale benchmarks.

1 INTRODUCTION

One of the major drawbacks of artificial intelligence compared to human intelligence is the lack of adaptivity to distributional shifts. While humans easily make correct decisions even on unseen domains, deep neural networks often exhibit significant performance degradation on the data in 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 kinds of domains and follows the true data distribution. Therefore, learning domain-invariant representations using limited data in source domains is critical to 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 make the model exposed to various domains during the training procedure. To stretch the coverage of the source domains, recent approaches often employ data generation strategies (Shankar et al., 2018; Zhou et al., 2020b;a; 2021b; Li et al., 2021; Nuriel et al., 2021; Yang et al., 2021; Zhong et al., 2022). While they have shown promising results on generalization ability, many of them require additional information of data such as domain labels for individual instances (Shankar et al., 2018; Zhou et al., 2020b;a; 2021b) or even extra network components such as generators and domain classifiers (Shankar et al., 2018; Zhou et al., 2020b;a; Yang et al., 2021). 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 free from extra information about data or additional network modules (Li et al., 2021; Nuriel et al., 2021), but they are limited to straightforward feature augmentations by adding trivial noise stochastically.

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 for instance normalization, which corresponds to the style of an input image. Based on this observation, existing works attempt to learn style-agnostic networks robust to domain shifts via style augmentations (Zhou et al., 2021b; Nuriel et al., 2021; Kang et al., 2022). Although they do not require additional networks (Zhou et al., 2021b; Nuriel et al., 2021), they are limited to using simple augmentation techniques with no feedback loop in the augmentation process, leading to suboptimal performance. While StyleNeophile (Kang et al., 2022) augments novel styles with different distributions using the information observed in the previous iterations, there is no guarantee of their positive behavior for unseen styles in the target domain.

Although our approach follows the same assumption as (Zhou et al., 2021b; Nuriel et al., 2021; Kang et al., 2022), its objective for style augmentation is to synthesize hard examples to improve trained models. To this end, inspired by adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018; Xie et al., 2019), the proposed method, referred to as Randomized Adversarial Style Perturbations (RASP), adversarially attacks the styles of features so that the corresponding examples are to be classified into the target labels different from the ground-truths. Unlike the other methods based on adversarial attacks toward the fixed target label (Shankar et al., 2018; Zhong et al., 2022), we employ randomized targeted attacks to diversify augmentation directions. While the features with modified styles strengthen the generalization ability on unseen domains, they might neglect crucial information observable in source domains since the style augmentation is prone to increase the output discrepancy even with perturbation-free features. To compensate for this, we propose Normalized Feature Mixup (NFM) technique based on Mixup (Zhang et al., 2018). 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 the Mixup with augmented styles, we successfully maintain the representations from the source domain while taking advantage of style augmentations 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 attacks, which is free from any architectural modifications or the need for the domain label of each example.
- We propose a novel feature Mixup method, NFM, which allows us to maintain knowledge from the source domains while facilitating the adaptation to fresh data via robust domain augmentation.
- The proposed approach consistently demonstrates the outstanding generalization ability in the 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 and adversarial attack 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

2.1 Domain Generalization

Domain Generalization (DG) aims to obtain generalization ability on unseen target domains while using only source domains during training. The existing line of work (Li et al., 2018; Balaji et al., 2018; Du et al., 2020) formulates the domain generalization task as a meta-learning problem by splitting source domains into the meta-train and meta-test sets. Adopting the learn-to-learn scheme of the meta-learning framework, they seek to learn to generalize on unseen domains. Other works (Seo et al., 2020; Fan et al., 2021) focus on the observation that the naive batch normalization technique that ignores domain-specific characteristics can hurt the generalization ability on unseen domains.

Recent studies on DG (Zhou et al., 2020a;b; Shankar et al., 2018; Zhou et al., 2021b; Nuriel et al., 2021; Li et al., 2021; Xu et al., 2021; Yang et al., 2021; Kang et al., 2022) investigate the methods

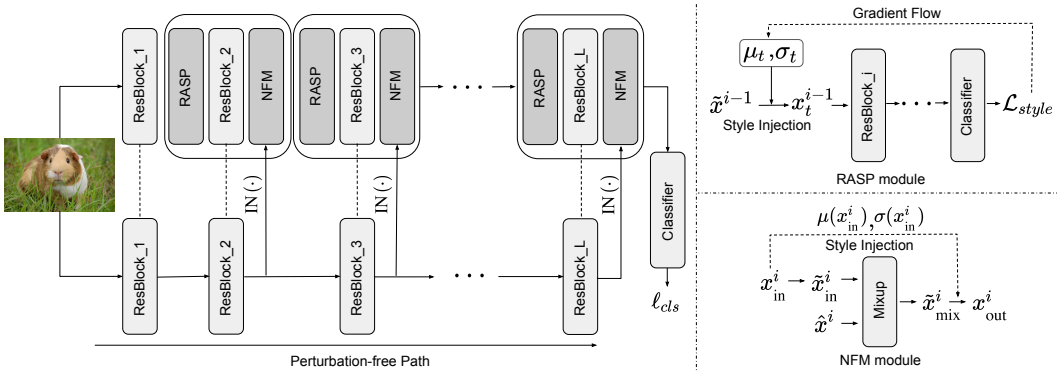

Figure 1: Overall framework of Randomized Adversarial Style Perturbations. Our model runs parallel paths one with our proposed Randomized Adversarial Style Perturbations (RASP), Normalized Feature Mixup (NFM) modules and one without them. Before each block, we apply RASP module to augment the novel styles. RASP adjust styles of each features by minimizing loss with respect to random target class different from ground truth. After passing RASP augmented features through the next block, we apply NFM to boost the learning of original features by utilizing perturbation-free path features while preserving effects of novel style augmentation by RASP. Note that weights of two paths are shared.

based on data generation and obtain domain-invariant feature extractors by introducing sufficient quantities of novel domain data at training time. A line of data generation based methods (Zhou et al., 2020b;a; Yang et al., 2021) rely on generator architecture which induces additional training cost and instability. While FACT (Xu et al., 2021) and CrossGrad (Shankar et al., 2018) do not use generator architecture, they deploy auxiliary neural networks including domain classifier and mean teacher network (Tarvainen & Valpola, 2017) to regularize the augmentation directions.

Others (Zhou et al., 2021b; Nuriel et al., 2021; Kang et al., 2022) focus on the observation that the feature statistics capture characteristics of the visual domains. MixStyle (Zhou et al., 2021b) generates novel domain features by mixing feature statistics of different images while pAdaIN (Nuriel et al., 2021) randomly permutes the statistics of each feature before every batch normalization layer. Similar to (Zhou et al., 2021b; Nuriel et al., 2021), SFA Li et al. (2021) adopts feature-level augmentation technique that applies simple stochastic rescaling and additive operation. Different form (Zhou et al., 2021b; Nuriel et al., 2021), StyleNeophile (Kang et al., 2022) augments styles that have different distribution from that of previous iterations. While StyleNeophile (Kang et al., 2022) does not depends on simple stochasticity, there is no guarantee that the augmented styles will be useful for generalization in target task.

2.2 Adversarial Attacks

Recently, the techniques that attempt to mislead the network output by injecting imperceptible noise into the data have been actively studied in the name of adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018; Xie et al., 2019; Zhao et al., 2018; Moosavi-Dezfooli et al., 2016). Many adversarial attack methods (Goodfellow et al., 2015; Madry et al., 2018; Xie et al., 2019) rely on gradient-based optimization. These methods generate adversarial noise by solving an optimization problem that aims to increase the loss corresponding to the ground-truth label.

3 Proposed Approach

This section describes the technical details of the proposed Randomized Adversarial Style Perturbations (RASP) and the Normalized Feature Mixup (NFM) methods.

3.1 Background

Instance normalization and styles Recent studies on neural style transfer (Huang & Belongie, 2017; Ulyanov et al., 2016) found that the style of an image can be captured by the instance-specific

channel-wise mean and standard deviation. Instance Normalization (IN) (Ulyanov et al., 2016) removes the effect of styles on images by normalizing features as follows:

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

where γ and β are learnable affine parameters and $(\mu(x), \sigma(x))$ is instance-specific channel-wise statistics computed by

$$\mu(x) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{h,w} \quad \text{and} \quad \sigma(x) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_{h,w} - \mu(x))^2}. \tag{2}$$

AdaIN (Huang & Belongie, 2017) allows the style of an input image y to be transferred to the content of another input x by replacing the affine parameters of IN with feature statistics of the style input y as follows:

$$\text{AdaIN}(x) = \sigma(y) \frac{x - \mu(x)}{\sigma(x)} + \mu(y). \tag{3}$$

Gradient-based adversarial attacks Adversarial attacks (Kurakin et al., 2016; Zhao et al., 2018; Moosavi-Dezfooli et al., 2016; Xie et al., 2019; Madry et al., 2018; Goodfellow et al., 2015) trick a neural network by injecting imperceptible small noise to an example. It has been actively explored in recent years, and 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 (Goodfellow et al., 2015) computes the optimal perturbation of a linearized cost function, which is given by

$$x = x + \epsilon \, \text{sign}(\nabla_x J_\theta(x, y)), \tag{4}$$

where ϵ is the magnitude of perturbation and $J_\theta(\cdot, \cdot)$ is a task-specific loss function. I-FGSM (Kurakin et al., 2016) extends FGSM (Goodfellow et al., 2015) by using multiple iterations as

$$x_0 = x \tag{5}$$

$$x_{t+1} = \text{clip}_{x,\epsilon} \left[x_t + \alpha \, \text{sign}(\nabla_x J_\theta(x_t, y)) \right], \tag{6}$$

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

3.2 Overall Framework

Let (x, y) denote an image and class label pair from an arbitrary source domain. We adopt Empirical Risk Minimization (ERM) using the standard classification loss denoted by ℓ_{cls} 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 employ a parallel forwarding path with shared weights called a perturbation-free path and make two paths interact with each other.

To enhance the generalization ability on unseen visual domains, we incorporate the Randomized Adversarial Style Perturbations (RASP) module, which will be discussed in Section 3.3, before each block with a probability of 0.5. Although adversarial style augmentation is effective to generate challenging styles, we also perform Normalized Feature Mixup (NFM) after each block with a probability of 0.5, when RASP is applied, to prevent the augmented styles 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 goal of RASP is to augment the styles of features in each block and enhance the robustness of the features regardless of their styles. We achieve the objective by transforming features using the styles that have the potential to mislead the network outputs at training time. While the existing style perturbation methods (Zhou et al., 2021b; Nuriel et al., 2021; Kang et al., 2022) are helpful for learning style-agnostic feature extractors, we argue that the target direction of style augmentation is

Algorithm 1 Randomized Adversarial Style Perturbations (RASP): before the i^{th} block

Input: feature extractor $f_\theta^L, \cdots, f_\theta^{i+1}, f_\theta^i$, classifier g_ϕ, number of iterations T, step size ϵ, target label y_{target}, ground-truth label y, feature from $(i-1)^{\text{st}}$ block x^{i-1}, threshold τ
$\mu_0 = \mu(x^{i-1}),\ \sigma_0 = \sigma(x^{i-1})$
$\tilde{x}^{i-1} = \dfrac{x^{i-1} - \mu_0}{\sigma_0}$
for $t = 0, \cdots, T-1$ **do**
    $x_t^{i-1} = \tilde{x}^{i-1} \cdot \sigma_t + \mu_t$
    $z_t = g_\phi \circ f_\theta^L \circ \cdots \circ f_\theta^i(x_t^{i-1})$
    if $\text{softmax}(z_t)_y \geq \tau$ **then**
        $\mathcal{L}_{\text{style}} = \ell_{\text{cls}}(z_t, y_{\text{target}})$
        $\mu_{t+1} = \mu_t - \epsilon \cdot \|\mu_0\|_2 \cdot \text{sign}(\nabla_{\mu_t} \mathcal{L}_{\text{style}})$
        $\sigma_{t+1} = \sigma_t - \epsilon \cdot \|\sigma_0\|_2 \cdot \text{sign}(\nabla_{\sigma_t} \mathcal{L}_{\text{style}})$
    else
        $\mu_T = \mu_t,\ \sigma_T = \sigma_t$
        break
    end if
end for
Output: $x_T^{i-1} = \tilde{x}^{i-1} \cdot \sigma_T + \mu_T$

essential since we aim to employ new styles effective for learning domain-agnostic models with a limited number of style-augmented examples instead of simply populating diverse but pointless data.

The style augmentation given by RASP has three characteristics, which are prediction disturbance, style diversity, and plausibility. Prediction disturbance reflects our main assumption that adversarial style augmentation prevents a network from being deceived by style changes in unseen target domains. Although attaining prediction disturbance, attacks towards predetermined targets may result in weak diversity of augmented styles. Hence, to resolve this issue, we introduce a randomized target selection strategy that guarantees the diversity of generated styles. For plausibility, we employ a threshold τ to terminate the RASP process when the confidence of the ground-truth label falls below the threshold; it ensures augmented styles are within realistic ranges.

To be specific, given a feature x^{i-1} after the $(i-1)^{\text{st}}$ block and a randomly sampled target label y_{target} different from the original label, we compute the style loss $\mathcal{L}_{\text{style}}$ at each attack iteration t as

$$\mathcal{L}_{\text{style}} = \ell_{\text{cls}}\left(g_\phi \circ f_\theta^L \circ \cdots \circ f_\theta^i(x_t^{i-1}), y_{\text{target}}\right), \tag{7}$$

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

$$\mu_{t+1} = \mu_t - \epsilon \cdot \|\mu_0\|_2 \cdot \text{sign}(\nabla_{\mu_t} \mathcal{L}_{\text{style}}) \tag{8}$$

$$\sigma_{t+1} = \sigma_t - \epsilon \cdot \|\sigma_0\|_2 \cdot \text{sign}(\nabla_{\sigma_t} \mathcal{L}_{\text{style}}), \tag{9}$$

as long as the following condition is met:

$$\text{softmax}(g_\phi \circ f_\theta^L \circ \cdots \circ f_\theta^i(x_t))_y \geq \tau. \tag{10}$$

Note that the ratio of the distribution shift on style in RASP tends to be preserved by scaling the learning rate using the norm of mean or standard deviation as in (9). Algorithm 1 presents the detailed procedure and mathematical formulation of RASP.

3.4 Normalized Feature Mixup (NFM)

Although RASP provides plentiful novel styles that strengthen generalization performance on unseen domains, it degrades the capability of learning features from source domains since perturbed styles may excessively diverge from the styles without perturbations. To make up for this phenomenon,

we propose Normalized Feature Mixup technique (NFM) that ensembles instance normalized features from both the perturbation-free path and the RASP path. By doing this, NFM preserves the knowledge from source domains while learning robust representations by taking advantage of style augmentations.

For NFM, instead of using the features that may change the augmented styles back into the original one, we perform Mixup with the normalized feature and integrate augmented style into the mixed normalized features. Given feature from the i^{th} block x^i_{in}, we obtain a feature after instance normalization with augmented style in the RASP path, $\tilde{x}^i_{\text{in}}$. NFM module mixes $\tilde{x}^i_{\text{in}}$ with instance-normalized feature from the perturbation-free path in the i^{th} block, $\hat{x}^i$, to get mixed normalized feature, $\tilde{x}^i_{\text{mix}}$, and denormalize it with the augmented style $(\mu(x^i_{\text{in}}), \sigma(x^i_{\text{in}}))$ to obtain the final output, x^i_{out}, as follows:

$$\tilde{x}^i_{\text{mix}} = \alpha \cdot \hat{x}^i + (1 - \alpha) \cdot \tilde{x}^i_{\text{in}}, \tag{11}$$

$$x^i_{\text{out}} = \tilde{x}^i_{\text{mix}} \cdot \sigma(x^i_{\text{in}}) + \mu(x^i_{\text{in}}), \tag{12}$$

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 use the perturbation-free path for inference and its computational complexity depends on the baseline algorithm.

4 Experiments

We demonstrate the performance of the proposed algorithm on the standard benchmarks for domain generalization and analyze the characteristics of our approach in comparison with existing techniques.

4.1 Datasets and Evaluation Protocol

We evaluate the proposed algorithm on DomainNet (Peng et al., 2019), Office-Home (Venkateswara et al., 2017), and PACS (Li et al., 2017), which are standard benchmarks for domain generalization. Our primary target benchmark is DomainNet because it is a large-scale dataset in terms of the number of classes and examples; it contains 586,475 images of 6 domains (Clipart, Infograph, Painting, Quickdraw, Real, 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, Sketches) and 7 classes (dog, elephant, giraffe, guitar, horse, house and person).

For evaluation, we follow the same leave-one-domain-out protocols as previous works (Zhou et al., 2021b; Huang et al., 2020; Zhou et al., 2020a). We select the final model using the validation set from all source domains. All the results are average classification accuracy over five runs with different random seeds.

4.2 Implementation Details

We employ ResNet18 and ResNet50 pre-trained on ImageNet (Deng et al., 2009) 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 epoch 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 $2/255$, and the number of attack iterations to 5 for all datasets. RASP and NFM are applied to the 2nd, 3rd, and 4th residual blocks, before and after each block, respectively. All of the experiments are performed on a single TITAN-XP GPU.

4.3 Results on DomainNet

Quantitative results on DomainNet with RASP and NFM are presented in Table 1. DomainNet is the most challenging dataset for DG since it is the largest dataset and there is a large gap between the domains it contains compared to other datasets. Despite these difficulties, the proposed algorithm

Table 1: Performance on DomainNet (Peng et al., 2019). RASP presents outstanding accuracy in this large-scale dataset. The bold-faced numbers indicate the best performance.

(a) Results of ResNet18 on DomainNet

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ERM	56.6	18.4	45.3	12.5	57.9	38.8	38.3
MetaReg (Balaji et al., 2018)	53.7	21.1	45.3	10.6	**58.5**	42.3	38.6
DMG (Chattopadhyay et al., 2020)	60.1	18.8	44.5	14.2	54.7	41.7	39.0
StyleNeophile (Kang et al., 2022)	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	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ERM	64.0	23.6	51.0	13.1	64.5	47.8	44.0
MetaReg (Balaji et al., 2018)	59.8	25.6	50.2	11.5	**65.5**	50.1	43.6
DMG (Chattopadhyay et al., 2020)	65.2	22.2	50.0	15.7	59.6	49.0	43.6
StyleNeophile (Kang et al., 2022)	66.1	21.4	51.4	15.3	61.7	51.8	44.6
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**

Table 2: Performance on Office-Home (Venkateswara et al., 2017). Note that the shaded rows are for the algorithms that use different hyperparameters for individual target domains. The bold-faced numbers indicate the best performance among the methods evaluated without domain-specific hyperparameters.

(a) Results of ResNet18 on Office-Home

Method	Art	Clipart	Product	Real	Avg.
ERM	59.0	48.4	72.5	75.5	63.9
CrossGrad (Shankar et al., 2018)	58.4	49.4	73.9	75.8	64.4
MixStyle (Zhou et al., 2021b)	58.7	53.4	74.2	75.9	65.5
StyleNeophile (Kang et al., 2022)	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 (Zhou et al., 2020a)	59.2	52.3	74.6	76.0	65.5
ADVTSRL (Yang et al., 2021)	60.7	52.9	75.8	77.2	66.7

(b) Results of ResNet50 on Office-Home

Method	Art	Clipart	Product	Real	Avg.
ERM	64.7	58.8	77.9	79.0	70.1
CrossGrad (Shankar et al., 2018)	67.7	57.7	79.1	80.4	71.2
MixStyle (Zhou et al., 2021b)	64.9	58.8	78.3	78.7	70.2
RASP+NFM (ours)	**68.8**± **0.4**	**61.7**± **1.0**	**79.8**± **0.2**	**80.9**± **0.3**	**72.8**
DDAIG (Zhou et al., 2020a)	65.2	59.2	77.7	76.7	69.7
ADVTSRL (Yang et al., 2021)	69.3	60.1	81.5	82.1	73.3

shows significant improvements in averaged accuracy with ResNet18 and ResNet50. As depicted in Table 1, our method boosts baseline (ERM) with a remarkable margin in all domains except Real domain with both ResNet18 and ResNet50. Not only improving the baseline, but our method also shows better performance than existing methods. Table 1 demonstrates that improvements in challenging domains (Infograph, Painting, Quickdraw, Sketch) are outstanding compared to easy domains (Clipart, Real). This illustrates that the proposed method, optimizing worst-case styles, can achieve more significant improvements when there is a large domain gap between source domains and target domains.

4.4 Results on Office-Home

To validate the effectiveness of our algorithm, we conducted experiments on Office-Home. Table 2 clearly shows that the proposed algorithm achieves better accuracy than existing methods on Office-Home. Similar to DomainNet, the proposed method exhibits stronger performance improvement in difficult target domains. This is consistent with the observation in DomainNet that RASP is more useful when there exists a larger discrepancy between source domains and target domains. Note that ADVTSRL (Yang et al., 2021) and DDAIG (Zhou et al., 2020a) adopt different hyper-parameters for each target domain, which makes direct comparison with those methods unfair.

Table 3: DG performance on ResNet18 for our model and the existing works on PACS (Li et al., 2017) Note that the shaded rows of the table denote the algorithms that use different hyper-parameters for each target domain. The bold-faced numbers indicate the best performance among the methods that do not require target-domain-specific hyper-parameters.

Results of ResNet18 on PACS

Method	Art	Cartoon	Photo	Sketch	Avg.
ERM	75.1	74.2	95.6	68.4	78.3
CrossGrad (Shankar et al., 2018)	79.8.	76.8	96.0	70.2	80.7
MixStyle (Zhou et al., 2021b)	84.1	78.8	**96.1**	75.9	83.7
StyleNeophile (Kang et al., 2022)	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 (Zhou et al., 2020a)	84.2	78.1	95.3	74.7	83.1
ADVTSRL (Yang et al., 2021)	85.8	80.7	97.3	77.3	85.3

Table 4: Ablation study results on the variations of our algorithm. The bold-faced numbers indicate the best performance.

Ablation types	Variations	Art	Cartoon	Photo	Sketch	Avg.
(a) Augmentation Objective	Ours	**84.6**	**79.8**	**94.1**	80.1	**84.7**
	Ground-truth	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**
(c) Attack Iterations (T)	3	83.6	78.8	**94.9**	77.6	83.7
	4	84.4	79.7	94.5	78.4	84.3
	5	**84.6**	79.8	94.1	80.1	**84.7**
	6	84.2	**80.0**	93.6	**80.4**	84.5
(d) Thresholding (τ)	0.0	83.0	79.0	92.1	77.7	83.0
	0.2	83.1	79.0	93.1	78.5	83.4
	0.4	83.5	79.6	93.0	**81.0**	84.3
	0.6	84.3	**79.8**	93.3	80.8	84.6
	0.8	**84.6**	**79.8**	**94.1**	80.1	**84.7**

4.5 Results on PACS

We evaluate our proposed modules RASP and NFM on PACS dataset. Table 3 presents the results on PACS. While the most of methods utilize domain labels or additional architectures, our method outperforms most of the existing methods in most of the domains. It can be seen that the performance gain from the proposed method is most significant in the most challenging domains including Sketch and Cartoon. On the other hand, in the case of the easy domains including Art painting and Photo, it is possible to learn useful information for the target domains only by learning from the source domains. Therefore, in this case, the role of NFM that preserves the important information from source domains becomes a critical issue. A detailed analysis of the role of NFM modules can be found in Section 4.6. Note that ADVTSRL (Yang et al., 2021) and DDAIG (Zhou et al., 2020a) adopt different hyper-parameters for each target domain, which makes direct comparison with those methods unfair.

4.6 Ablation Studies

To validate our algorithm, we perform various ablation studies, which include the analysis of hyper-parameters, objective functions, and implementation variations. All of the experiments were done in ResNet18 on PACS.

Effects of attack objectives To show that our choice of augmentation direction is optimal, we investigate the results with varying augmentation directions. One simple way to achieve prediction disturbance might be increasing the classification loss with respect to the ground truth label. As we can see in Table 4 (a), such direction clearly shows inferior results to the proposed direction, which is increasing loss with respect to the randomly sampled target classes. We conjecture that this

Table 5: Ablation study results on the location our method is applied. The bold-faced numbers indicate the best performance.

RGB	Res2	Res3	Res4	Art	Cartoon	Photo	Sketch	Avg.
				75.1	74.2	95.6	68.4	78.3
✓				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**

phenomenon occurs because randomly selected target classes guarantee more diverse augmentation in the style space.

Effects of variation on NFM Table 4 (b) presents the results with variations on NFM modules. When Mixup is applied to the styles or features instead of normalized features, classification accuracy significantly drops. We infer this is due to such Mixup strategies weaken the advantages of the RASP that provide novel styles by integrating normal styles into the augmented styles.

Effects of attack iterations To further investigate the effects of RASP, we analyze the influence of attack iteration T. Table 4 (c) shows the evaluation results on PACS with varying attack iteration. As more iterations produce more challenging styles for networks, we can see that classification accuracy on the comparably easy target domain, *e.g.,* photo, decreases as iteration increases while the classification accuracy in the comparably difficult target domain, e.g. sketch, increases as iteration increases.

Effects of thresholding We evaluate our proposed algorithms with different thresholds. Table 4 (d) shows the varying results with different thresholds. The threshold is a hyper-parameter that balances the diversity and plausibility of the synthesized styles. By avoiding the generation of implausible styles, large thresholds aid in the learning of sufficient information from the source domain in a situation with a relatively small domain gap. On the other hand, if the domain gap is substantial, small thresholds can improve performance by enabling the generation of novel styles. Conversely, if the threshold is too low, it may degrade performance by generating harmfully diverse styles.

Effects on where to apply the proposed algorithm We evaluate our proposed algorithms multiple times while applying our method to different levels of the network. Table 5 shows that the performance of the proposed method varies greatly depending on the location where it is applied. In particular, it can be seen that when the proposed module is applied to the very early or late blocks, the performance improvement is negligible compared to when it is applied to the middle-level blocks.

5 Conclusion

We presented a simple yet effective style augmentation framework for domain generalization which is based on adversarial attacks. The proposed method augments the styles that can deceive the network by attacking the model itself. The augmented styles achieve the generalization ability on unseen domains by making feature extractors robust to the style changes that might mislead the network outputs. Proposed RASP and NFM, since they do not require any architectural modifications or domain labels, can be easily attached to the existing baselines. The extensive experimental shows that the proposed algorithm consistently improves generalization ability on unseen target domains on multiple datasets.

5.1 Limitations

While the proposed algorithm does not require additional architectures, it also requires additional computations to augment the styles of the features compared to vanilla training of convolutional neural networks at training time.

Ethics Statement Domain Generalization is related with fairness issue since neural architectures without domain generalization ability will make unexpected decisions, which might be unfair decisions, on unseen domains. By achieving domain generalization, our method contributes to resolving in-nature unfairness of deep learning models.

Reproducibility We present detailed procedure of the proposed algorithm in Section 3. Following the procedure, the results can be reproduced by using hyper-parameters in Section 4.2 We will release our codes soon.

REFERENCES

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

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

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.

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.

Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *CVPR*, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

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

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020.

Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *CVPR*, 2022.

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

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

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

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

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

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

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

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

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.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

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

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

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

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.

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

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

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

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

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

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020a.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020b.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing (TIP)*, 2021a.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021b.