On the Fly Neural Style Smoothing for Risk-Averse Domain Generalization

Akshay Mehra[1], Yunbei Zhang[1], Bhavya Kailkhura[2], and Jihun Hamm[1]
[1]Tulane University [2]Lawrence Livermore National Laboratory
{amehra, yzhang111, jhamm3}@tulane.edu, kailkhura1@llnl.gov

Abstract

Achieving high accuracy on data from domains unseen during training is a fundamental challenge in domain generalization (DG). While state-of-the-art (SOTA) DG classifiers have demonstrated impressive performance across various tasks, they have shown a bias towards domain-dependent information, such as image styles, rather than domain-invariant information, such as image content. This bias renders them unreliable for deployment in risk-sensitive scenarios such as autonomous driving where a misclassification could lead to catastrophic consequences. To enable risk-averse predictions from a DG classifier, we propose a novel inference procedure, Test-Time Neural Style Smoothing (TT-NSS), that uses a "style-smoothed" version of the DG classifier for prediction at test time. Specifically, the style-smoothed classifier classifies a test image as the most probable class predicted by the DG classifier on random re-stylizations of the test image. TT-NSS uses a neural style transfer module to stylize a test image on the fly, requires only black-box access to the DG classifier, and crucially, abstains when predictions of the DG classifier on the stylized test images lack consensus. Additionally, we propose a neural style smoothing (NSS) based training procedure that can be seamlessly integrated with existing DG methods. This procedure enhances prediction consistency, improving the performance of TT-NSS on non-abstained samples. Our empirical results demonstrate the effectiveness of TT-NSS and NSS at producing and improving risk-averse predictions on unseen domains from DG classifiers trained with SOTA training methods on various benchmark datasets and their variations.

1. Introduction

The objective of Domain Generalization (DG) [75] is to develop models that demonstrate remarkable resilience to domain shifts during testing, even without prior knowledge of the test domain during training This represents a challenging problem, as it is impractical to train a model to be robust to all potential variations that may arise at test time. For example, previous works [2, 7, 11, 27, 30] have demonstrated that variations in styles/textures, weather changes, etc., unseen during training can drastically reduce the classifier's performance. Recent works [5, 27, 35, 56] brought to light the fact that predictions from state-of-the-art (SOTA) neural networks are biased towards the information unrelated to the content of the images but are dependent on the image styles, a characteristic that can vary across domains. Due to the vast practical implications of this problem many works have studied this problem both analytically [8–10, 41, 51, 53, 62, 84] and empirically [1,24,28,54,59,78,85]. However, in scenarios such as in autonomous driving, medical diagnoses, or rescue operations involving drones, where misclassifications can have severe consequences, it becomes essential to augment classifiers with abstaining mechanisms or involve humans in the decision-making process [19,61]. In this work, we focus on the problem of image classification under distribution shifts which comprise of differences in image styles.

To safeguard the classifier against risky misclassification (and enable risk-averse predictions) we augment the classifier with a capability to defer making a prediction on samples, when it lacks confidence. However, since the softmax score of the classifier is known to be uncalibrated [29,32,34] on data from unseen domains, we propose a novel test-time method that uses neural style information to estimate classifier's confidence in its prediction under style changes. Our inference procedure, Test-Time Neural Style Smoothing (TT-NSS), depicted in Fig. 1, first transforms a classifier (base classifier) into a style-smoothed classifier and then uses it to either predict the label of an incoming test sample or abstain on it. Specifically, the prediction of the style smoothed classifier, ψ, constructed from a base classifier f, on a test input x is defined as the class that the base classifier f predicts most frequently on stylized versions of the input. TT-NSS uses a style transfer network based on AdaIN [36] to produce stylized versions of the test input in real-time. While AdaIN can transform the style of x to any arbitrary style, we specifically transform it into the style of the data from the domains used for training. This choice is based on the assumption that f can be made agnostic to the

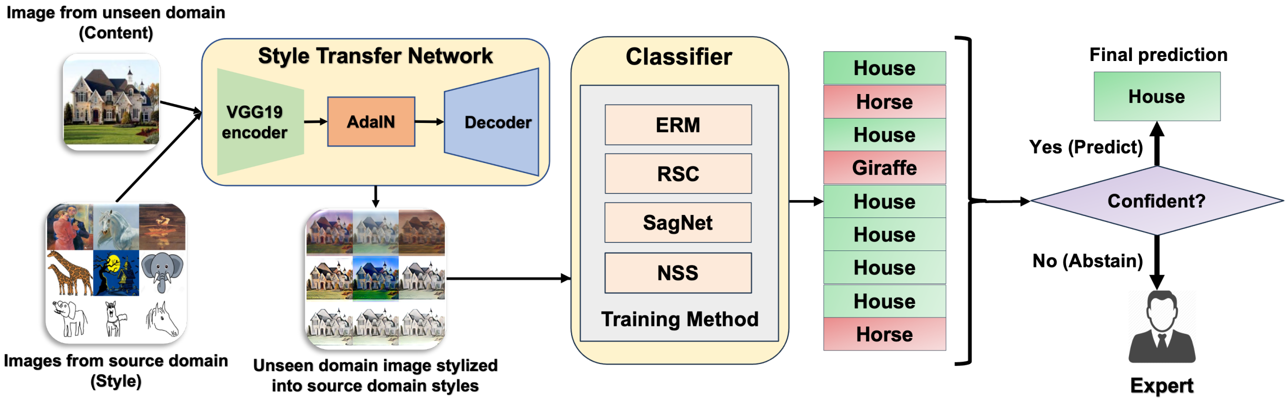


Figure 1. Overview of our Test-Time Neural Style Smoothing (TT-NSS) inference procedure for obtaining risk-averse predictions. TT-NSS works by stylizing a test sample into source domain styles and classifies the sample as the most probable class assigned by the base DG classifier to the stylized samples if that class is much more likely than the other classes. Otherwise, it abstains from making a prediction and refers the sample to an expert thereby avoiding a risky misclassification.

styles of the data from domains used for training. Moreover, changing the styles of x to arbitrary styles, unknown to f, can worsen the classifier's performance due to a widened distribution shift.

TT-NSS can be used to evaluate any DG classifier with only black-box access to it, i.e., it does not require the knowledge of weights, architecture, or training procedure used to train the classifier and only needs its predictions on stylized test samples. However, computing the prediction of a style-smoothed classifier requires computing the probability with which the base classifier classifies the stylized images of x. Following works in Randomized Smoothing [18], we propose a Monte Carlo algorithm to estimate this probability. When this estimated probability exceeds a set threshold it implies that the predictions of the classifier f on stylized images of x achieve a desired level of consensus and the prediction is reliable. In other cases, TT-NSS abstains due to a lack of consensus among the predictions of the base DG classifier. Recently, test-time adaptation [39, 83] (TTA) approaches have been shown to be effective in the DG setup which adapts some or all parameters of the classifier using multiple incoming data samples from the unseen domains. However, our work differs significantly from these since we consider a black-box setting where parameters of the classifier are not accessible at test time making our approach much more practically useful compared to TTA approaches.

Furthermore, we propose a novel training procedure based on neural style smoothing (NSS) to improve the consistency of the predictions of the DG classifier on stylized images. The improved consistency leads to improved performance of the DG classifier on non-abstained samples at lower abstaining rates making them more reliable. Our training method creates a style-smoothed version of the soft base DG classifier and uses stylized versions of the source domain data (generated by stylizing the source domain images into random styles of other source domain images) to train the base DG classifier. Similar to previous works [40, 65, 66], we incorporate consistency regularization during training to further boost the performance of the classifier on non-abstained samples at various abstaining rates. Similar to TT-NSS which can be used with any classifier, our NSS-based training losses can be combined with any training method and can help improve the reliability of the classifier's predictions without significantly degrading their accuracy or requiring access to auxiliary data from unseen domains [16, 33]. We present results of using our inference and training procedures on PACS [47], VLCS [22], Office-Home [72] and their variations generated by applying style changes and common corruptions, in both single and multiple source domain settings. Our results show the effectiveness of our proposed methods at enabling and improving risk-averse predictions from classifiers trained with SOTA DG methods on data from unseen domains. Our main contributions are summarized below:

- We focus on the problem of obtaining risk-averse predictions in a DG setup with black-box access to the classifier. We propose an efficient inference procedure relying on AdaIN-based style transfer and a style-smoothed classifier for classification and abstaining.
- To improve the quality of risk-averse predictions, we propose losses that enforce prediction consistency on the random stylization of the source data and can be seamlessly combined with losses of any DG method.
- We demonstrate the effectiveness of our inference and training methods on benchmark datasets and their variations generated by stylizing and using corruptions.

2. Related work

Domain generalization: The goal of domain generalization (DG) is to produce classifiers whose accuracy remains high when faced with data from domains unseen during training. Many works have proposed to address this problem by capturing invariances in the data by learning a representation space that reduces the divergence between multiple source domains thereby promoting the use of only domain invariant features for prediction [1,24,28,59,78,85]. Another line of work learns to disentangle the style and content information from the source domains and trains the classifier to be agnostic to the styles of the source domains [3,20,55,81]. Yet another line of research focuses on diversifying the source domain data to encompass possible variations that may be encountered at test time [12,34,44,66,74]. Unlike previous works which focus on improving classifier accuracy on unseen domains, we focus on making DG risk-averse on data from unseen domains.

Certified robustness via randomized smoothing: Many works have demonstrated the failure of SOTA machine learning classifiers on adversarial examples [14, 15, 38, 68, 77]. In response, many works proposed to provide empirical [4] and provable [18,45,46,52,60,80] robustness to these examples. Among them, Randomized Smoothing (RS) [18, 45, 46] is a popular method which considers a smoothed version of the original classifier and certifies that no adversarial perturbation exists within a certified radius (in ℓ_2 norm) around a test sample that can change the prediction of the classifier. RS uses Gaussian noise to produce a smoothed version of the base classifier and classifies a test sample to be the class most likely to be predicted by the base classifier on Gaussian perturbations of the test sample. While RS was proposed to certify the robustness to additive noise, the idea has been extended to certify robustness to parameterized transformations of the data such as geometric transformation [23, 48] where the noise is added to the parameters of the transformations. Our neural style smoothing procedure is similar to RS with crucial differences. Firstly, we use neural styles for smoothing (which cannot be parameterized) instead of adding Gaussian noise to the input or parameters of specific transformations. Secondly, our goal is not to provide certified robustness guarantees against style changes but to provide a practical method to produce reliable predictions on test samples and an abstaining mechanism to curb incorrect predictions.

Neural style transfer: Following [25], which demonstrated the effectiveness of using the convolutional layers of a convolutional neural network for style transfer, several ways have been proposed to improve style transfer [21, 26, 42, 70, 71, 76]. AdaIN [36] is a popular approach that allows style transfer by changing only the mean and variance of the convolutional feature maps. Other ways of generating stylized images include mixing [89] or exchanging [69, 86] styles, or using adversarial learning [63, 88].

Test-time adaptation (TTA): Recent works have demonstrated the effectiveness of using TTA for improving generalization to unseen domains, where the classifier is updated partially or fully using incoming batches of test samples [67, 73, 83]. This approach has also been shown to be effective in the DG setup [39]. Our approach is different from these methods since we do not assume access to the parameters of the DG classifier or assume that data from unseen domains arrive in batches.

Classification with abstaining: A learning framework allowing a classifier to abstain on samples has been studied extensively [6, 13, 17, 19, 57]. Two main approaches in these works include a confidence-based rejection where the classifier's confidence is used to abstain based on a predefined threshold and a classifier-rejector approach where the classifier and rejector are trained together. Our work is closer to the former since we do not train a rejector and abstain when the top class is not much more likely than other classes.

3. Neural style smoothing

3.1. Background

Domain Generalization (DG) setup: Given data samples $\mathcal{D}^i_{\text{source}} = \{(x^i_j, y^i_j)\}_{j=1}^{N^i}$, with N^i samples, from N_S source domains each following a distribution $P^i_S(X, Y)$, the goal of DG is to learn a classifier $f(X)$ whose performance does not degrade on a sample from an unseen test domain with distribution $P_T(X, Y) \neq P^i_S(X, Y)$, for all $i \in \{1, \cdots, N_S\}$. Depending on the number of source domains available during training the setup can be termed as single or multi-domain. The lack of information about the target domain makes the problem setup challenging and many previous works have proposed training methods focusing on capturing domain invariant information from source domain data to improve performance on unseen domains at test time. In the multi-domain setup, learning a classifier by minimizing its empirical risk on all available source domains achieves competitive performance on various benchmark datasets [28].

Neural style transfer with AdaIN [36]: Given a content image, x_c and a style image x_s, AdaIN generates an image having the content of x_c and style of x_s. AdaIN works by first extracting the intermediate features (output of `block4_conv1`) of the style and content image by passing them through a VGG-19 [64] encoder, g, pretrained on Imagenet. Using these features AdaIN aligns the mean (μ) and variance (σ) of the two feature maps using

$$\begin{aligned} t &= \mathrm{AdaIN}(g(x_c), g(x_s)) \\ &= \sigma(g(x_s)) \left(\frac{g(x_c) - \mu(g(x_c))}{\sigma(g(x_c))} \right) + \mu(g(x_s)). \end{aligned} \tag{1}$$

A decoder, h, is then used to map the AdaIN-generated feature back to the input space to produce a stylized image $x_{\text{stylized}} = h(t)$. We follow the design of the decoder as proposed in [36] and train the decoder to minimize the content loss between the features of the stylized image, $g(x_{\text{stylized}})$ and the AdaIN transformed features of the content image, i.e.

$$\mathcal{L}_{\text{content}} = \|g(x_{\text{stylized}}) - t\|_2^2, \tag{2}$$

along with a style loss that measures the distance between the feature statistics of the style and the stylized image using L layers of the pretrained VGG-19 network, ϕ. In particular, the style loss is computed as

$$\begin{aligned}\mathcal{L}_{\text{style}} = &\sum_{i=1}^{L} \|\mu(\phi_i(x_s)) - \mu(\phi_i(x_{stylized})\|_2^2 \\ &+ \sum_{i=1}^{L} \|\sigma(\phi_i(x_s)) - \sigma(\phi_i(x_{\text{stylized}})\|_2^2.\end{aligned} \tag{3}$$

We measure the style loss, using `block1_conv1`, `block2_conv1`, `block3_conv1`, and `block5_conv1` layers of the VGG-19 network. We pre-train the decoder with MS-COCO [49] images as content and Wikiart [58] images as style.

3.2. Neural style smoothing-based inference

Consider a classification problem from $\mathbb{R}^d$ to the label space $\mathcal{Y}$. Neural style smoothing produces an output, for a test image x, that a base DG classifier, $f : \mathbb{R}^d \to \mathcal{Y}$ is most likely to return when x is stylized into the style of the source domain data, i.e., the data used for training f. Formally, given a base DG classifier f, we construct a style-smoothed classifier $\psi : \mathbb{R}^d \to \mathcal{Y}$, whose prediction on a test image x is the most probable output of f on x converted into the style of the source domain data, i.e.,

$$\psi(x) := \arg\max_{y \in \mathcal{Y}} \mathbb{P}(f(h(t)) = y), \tag{4}$$

where $t = \text{AdaIN}(g(x), g(x_s))$, $x_s \sim P_S$, and P_S is the distribution of the source domain. When data from multiple source domains are available we combine the data from all the domains and use the combined data as source domain data. If the base DG classifier, f, correctly classifies the test image x when stylized into the styles of the source domain, then the style-smoothed classifier also correctly classifies that sample. However, computing the actual prediction of the style-smoothed classifier requires computing the exact probabilities with which the base DG classifier classifies the stylized test samples into each class. Thus, following [18], we propose a Monte Carlo algorithm to estimate these probabilities and the prediction of the style-smoothed classifier. The first step in estimating the prediction of the style-smoothed classifier on a test image x is to generate stylized versions of the image using the styles from the source domain. To achieve the style conversion in real-time, we use the AdaIN framework described previously with the content image as the test image x and n randomly chosen images from the dataset used for training the DG classifier as style images. The style transfer network then transforms x into n stylized images, each having the style of the source domain data, as illustrated in Fig. 1. The stylized images are then passed through the f and the class that is predicted the most often (majority class) is returned as the prediction of the test image. This procedure of Test-Time Neural Style Smoothing (TT-NSS) is detailed in Alg. 1.

Algorithm 1 Test-Time Neural Style Smoothing (TT-NSS)

Input: Test image x, base DG classifier f, VGG-19 encoder g, AdaIN decoder h, number of source style images n, $\mathcal{D}_{\text{styles}} = \{x_s^i\}_{i=1}^n$, threshold α.
Output: Prediction for x or ABSTAIN.

```
Initialize class-wise counts class_counts to zeros

# Generate $n$ stylized images from $x$ using $\mathcal{D}_{\text{styles}}$
for $i = 1, \cdots, n$ do
    $t = \text{AdaIN}(g(x), g(x_s^i))$
    $x_{\text{stylized}} = h(t)$
    prediction $= f(x_{\text{stylized}})$
    class_counts[prediction]$+= 1$
end for

# Get the top predicted class on stylized images
$c_{\max}$ = index of class_counts with highest count
$n_{\max}$ = class_counts[$c_{\max}$]

# Predict or ABSTAIN
if $\frac{n_{\max}}{n} < \alpha$ then
    return ABSTAIN
else
    return $c_{\max}$
end if
```

To ascertain that the prediction returned by TT-NSS is reliable, we estimate the confidence of the style-smoothed classifier in its prediction. In particular, we compute the proportion of the re-stylized test images that are classified as a particular class by the base DG classifier and obtain the counts of how often each class is predicted. Based on these counts, we compute the class which has the highest occurrence and if the proportion of the highest class exceeds a threshold α, TT-NSS classifies the test image as this class. However, if the proportion remains less than the threshold, then TT-NSS abstains due to a lack of consensus among the predictions. The abstained samples can then be sent for further processing to experts and save the system from returning a potentially incorrect prediction. A high value of α in

TT-NSS improves the accuracy on non-abstained samples but it also increases the number of abstained samples. On the other hand, a low value of α leads to decreased abstaining with an increased chance that the DG classifier may not be confident in its prediction, leading to a risky misclassification. In our empirical analysis in Sec. 4, we use various values of α ranging from 0 to 1 and show how the accuracy on non-abstained samples and the proportion of abstained samples change as the value of α is varied.

3.3. Neural style smoothing-based training

The performance of our inference procedure, TT-NSS, relies on the assumption that the base classifier, f, can classify the test image stylized into the source domain styles correctly and consistently. This requires that the base classifier be accurate on the images generated by the decoder used in the AdaIN-based neural style transfer network. However, our empirical evaluation of using TT-NSS on classifiers trained with existing DG methods on benchmark datasets shows a relatively low accuracy on non-abstained samples at smaller abstaining rates. This suggests that the base classifier cannot accurately classify the stylized images generated through the AdaIN decoder. Thus, we propose a new training procedure based on neural style smoothing (NSS) that enables consistent and accurate predictions from the classifiers when evaluated using TT-NSS. The proposed loss functions can be combined with any DG training algorithm and can be used to improve the reliability of the predictions from classifiers when evaluated with TT-NSS. To achieve this, we propose to augment the losses of an existing DG method with two additional loss functions. The first loss penalizes misclassification of the stylized images w.r.t. the label of the content image i.e., given a sample $(x, y) \sim \mathcal{D}_{\text{source}}$, the stylized misclassification loss is

$$\mathcal{L}_{stylized\_aug} = \mathbb{E}_{x_s \sim P_S}[\ell(f(h(t)), y)], \tag{5}$$

where $t = \text{AdaIN}(g(x), g(x_s))$ and ℓ is the cross entropy loss. Specifically, we first stylize a sample x from the source domain using multiple randomly sampled style images from the source domain and then penalize the misclassification loss of the classifier f on these stylized images. For a single source domain problem, even though all images from a domain may be considered as being in the same broad set of styles such as Art or Photos, individually the images have different non-semantic information such as textures, colors, patterns, etc., and thus stylizing an image into the styles of other source domain images is still effective and meaningful. The second loss which helps improve the trustworthiness of the predictions enforces consistency among the predictions of the stylized versions of the content image, generated using AdaIN. Previous works [40,65,66,87], have also demonstrated the effectiveness of enforcing consistency among the predictions of the classifier to be helpful in various setups such as semi-supervised learning and randomized smoothing. To define the style consistency loss, let $(x, y) \sim \mathcal{D}_{\text{source}}$, $F : \mathbb{R}^d \to \Delta^{K-1}$ be the softmax output of the classifier such that the prediction of the base classifier $f(x) = \arg\max_{k \in \mathcal{Y}} F(x)$, Δ^{K-1} be the probability simplex in $\mathbb{R}^K$, $\overline{F}(x) = \mathbb{E}_{x_s \sim P_S}[F(h(t))]$ with $t = \text{AdaIN}(g(x), g(x_s))$ be the average softmax output of the classifier on stylized images, $\text{KL}(\cdot\|\cdot)$ be the Kullback–Leibler divergence (KLD) [43] and $\text{H}(\cdot)$ be the entropy. Then the style consistency loss is given by

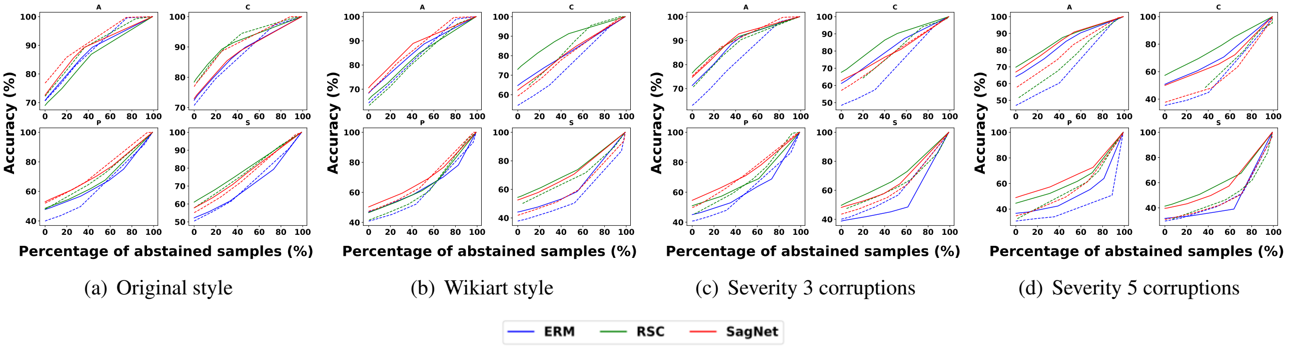
$$\begin{aligned}\mathcal{L}_{consistency} = \mathbb{E}_{x_s \sim P_S}[\text{KL}(\overline{F}(x) \| F(h(t)))] \\ + \text{H}(\overline{F}(x), y).\end{aligned} \tag{6}$$

In practice, we minimize the empirical version of the two losses using multiple-style images sampled randomly from the available source domain data. The trained classifier can then be evaluated using TT-NSS as in Alg. 1 to gauge the reliability of their predictions on unseen domains.

4. Experiments

In this section, we present the evaluation results of using our inference and training procedures for obtaining and improving the risk-averse predictions from DG classifiers. We present evaluations and comparisons with three popular DG methods, namely Empirical Risk Minimization (ERM), Style Agnostic Networks (SagNet), [56] and networks trained with Representation Self-Challenging (RSC) [37]. Our evaluation includes three popular benchmark datasets, namely PACS [47], VLCS [22] and OfficeHome [72], all of which contain four domains (see Appendix B). We also create and present evaluations on variations of these datasets generated by stylizing the images into the styles of Wikiart [58] and changing styles based on changes in weather, lighting, blurring, and addition of noise by using common corruptions [31] including {`frost, fog, brightness, contrast, gaussian blur, defocus blur, zoom blur, gaussian noise, shot noise, impulse noise`}. These variations allow us to evaluate the performance of DG classifiers on realistic changes that do not affect the semantic content of the images. To generate images from benchmark datasets stylized into the style of Wikiart, we use an AdaIN decoder pre-trained using images from MS-COCO [49] as content images and images from Wikiart [58] as style images. To create corrupted versions, we follow [31] and use corruption with severity levels 3 and 5. For reporting results over corrupted versions we use a subsample of the test set described in App. B.2 where as for original/wikiart styles we report results on the entire test set.

Following previous works [28], we used ResNet50 pretrained on the ImageNet dataset as our backbone network augmented with a fully connected layer with softmax activation. We use this network for training ERM and for neural



(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

Figure 2. Comparison of TT-NSS (solid lines) and confidence-based abstaining method (dashed lines) at producing risk-averse predictions in a **single** source domain setup on classifiers trained with SOTA DG methods. The graphs show accuracy vs abstained points on different variants of the **PACS** dataset ((a) original, (b) wikiart, (c,d) corrupted). In most domains, the accuracy of TT-NSS is higher than the corresponding accuracy of the confidence-based method for most of the range of the percentage of abstained samples demonstrating the superiority of TT-NSS at producing risk-averse predictions. (Note: The source domain from PACS used for training is denoted in the title.)

style smoothing (combined with ERM as the DG method). For other baselines, we train the classifiers using the source codes from the official repositories of RSC [37] and SagNet [56]. For all experiments in the single source domain setup, we train the classifiers with a single source domain and evaluate the performance of the remaining three domains. For multi-domain setup, we train the classifiers with three domains and test on the fourth unseen domain.

We compare the performance of TT-NSS (Alg. 1) with an abstaining mechanism that uses the classifier's max confidence on the original test sample for abstaining. In this method, we abstain if the highest softmax score for a sample is below a set threshold. We note that, compared to TT-NSS, which only requires prediction of the classifier on a sample the confidence-based mechanism additionally requires the classifier's confidence in the prediction and hence has access to more information than that available to TT-NSS, making TT-NSS more practically viable. For TT-NSS we use 10 randomly sampled style images ($n = 10$) for the single source domain setup and 15 for the multiple source domain setup (see Sec. 4.4). We present the accuracy of the DG classifier on non-abstained samples as a function of the proportion of abstained samples and the area under this curve (AUC) to demonstrate the effectiveness of TT-NSS (Alg. 1) and the confidence-based abstaining mechanism for producing risk-averse predictions. A higher AUC is desired since it indicates that the accuracy of the DG classifier at different abstaining rates remains high suggesting that whenever the inference procedure does not abstain, it is likely that the prediction is correct. This improves the reliability of the predictions from a DG classifier. We present additional experimental results in App. A followed by dataset and implementation details in App. B. Our codes are present at `https://github.com/akshaymehra24/RiskAverseDG`

4.1. TT-NSS improves the reliability of the predictions from existing DG classifiers

In this section, we demonstrate the effectiveness of TT-NSS at producing reliable predictions from classifiers trained with ERM, RSC, and SagNet when evaluated on domains unseen during training. The results in Fig. 2 and Figs. 7, 6 (in the Appendix) show the advantage of using the style-smoothed classifier over the confidence of the original classifier for producing risk-averse predictions on a test sample on PACS and VLCS datasets in both single and multiple source domain setting. This superiority of TT-NSS is also evident from the results in Tables 3, 5, 4, 6 (in the Appendix) which show the area under the curve for accuracy versus percentage of abstained samples for different settings. The high accuracy of the classifiers with TT-NSS at the same abstaining rates compared to the confidence-based strategy shows the advantage of TT-NSS at producing better risk-averse predictions. This advantage of TT-NSS becomes more apparent on stylized and corrupted variants of the PACS dataset where the standard accuracy of the classifier drops significantly and necessitates abstaining for safeguarding against risky misclassifications. The classifier's high confidence incorrect predictions on unseen domains is the primary reason that prevents the confidence-based strategy from producing risk-averse predictions. This is in line with the findings from previous works which have shown that a classifier can produce high-confidence misclassification on samples from unseen domains [29,32,50,79,82]. On the other hand, using the confidence of the style-smoothed classifier, by stylizing the test sample into source domain styles, can mitigate the classifier's bias to non-semantic information in the test samples and produce better quality pre-

Table 1. Effectiveness of NSS at producing a better AUC score compared to classifiers trained with ERM in a **single** source domain setting on PACS, VLCS, and OfficeHome datasets and their variations when evaluated with TT-NSS. The source domain used for training is denoted in the columns. (In all tables, the best result is marked in bold if the difference in the AUC is at least 0.01.)

	PACS				VLCS				OfficeHome			
Alg.	A	C	P	S	C	L	S	V	A	C	P	R
	Original Style											
ERM	0.875	0.878	0.662	0.702	0.567	**0.724**	0.851	0.751	0.689	0.553	0.549	0.685
NSS	**0.884**	**0.911**	**0.694**	**0.745**	**0.619**	0.685	0.853	**0.796**	**0.727**	**0.683**	**0.675**	**0.767**
	Wikiart Style											
ERM	0.854	0.816	0.643	0.626	0.477	0.682	0.785	0.704	0.552	0.344	0.321	0.5
NSS	0.855	**0.888**	**0.71**	**0.706**	**0.528**	**0.673**	**0.845**	**0.788**	**0.696**	**0.643**	**0.625**	**0.725**
	Corrupted with severity 3											
ERM	0.886	0.812	0.622	0.545	0.468	0.551	0.689	0.471	0.573	0.358	0.312	0.54
NSS	**0.901**	**0.853**	**0.717**	**0.683**	**0.573**	**0.686**	**0.775**	**0.608**	**0.625**	**0.576**	**0.56**	**0.67**
	Corrupted with severity 5											
ERM	0.834	0.708	0.519	0.468	0.411	0.439	0.567	0.415	0.445	0.235	0.196	0.383
NSS	**0.871**	**0.792**	**0.682**	**0.606**	**0.512**	**0.61**	**0.722**	**0.537**	**0.545**	**0.478**	**0.466**	**0.565**

dictions even without abstaining. This is evident from Fig. 2 and Figs. 7, 6 (in the Appendix) where TT-NSS (solid lines) achieve higher accuracy even at an abstaining rate of 0%.

Another crucial insight obtained from our evaluation on variations of benchmark datasets created by style changes is the significant decrease in the performance of the DG classifiers compared to the evaluation on original styles of the benchmark datasets both with confidence-based abstaining and TT-NSS. This suggests that classifiers trained with existing DG methods are susceptible to non-semantic variations in the data and improving the performance on these benchmark datasets while important may not be enough to achieve the goal of DG. However, while data augmentation and style diversification methods have been shown to be effective at improving the performance of DG methods on potential variations, it is not practical to train classifiers to be robust to all possible variations. Due to this limitation, improving the test time methods which either adapt the classifier to unseen domains or abstain from making predictions such as TT-NSS by explicitly transforming the test sample into known styles are essential for DG.

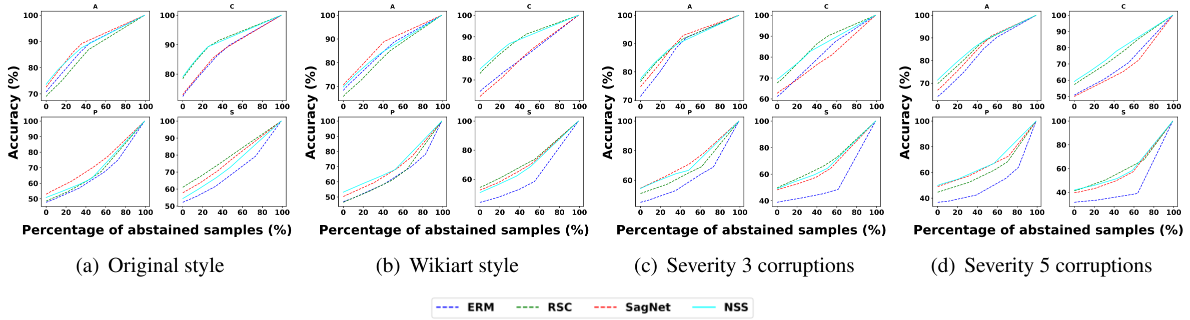
4.2. Effectiveness of NSS at improving risk-averse predictions from DG classifiers

Here we demonstrate the advantage of using the NSS training procedure for improving the reliability of the classifier's predictions. Specifically, we use the NSS losses with that of the ERM-based DG method and minimize the misclassification loss on source domain samples along with minimizing the style misclassification and style consistency losses. For training NSS with ERM we used four randomly sampled style images to compute the style smoothed losses in our experiments since we did not observe any significant performance difference with using more images. The use of a small number of style-transformed images during NSS training allows us to train DG classifiers without significantly increasing the computational cost compared to that of training with ERM. The stylized images were generated by using the AdaIN-based decoder pre-trained using data from MS-COCO [49] as content and Wikiart as style. Our results in Table 1 and Table 7 (in the Appendix) show that classifiers trained with NSS achieve a significantly better area under curve compared to classifiers trained with ERM on PACS, VLCS and OfficeHome datasets in both single and multiple source domain settings. The improvements in AUC become more evident on variations of these datasets generated by changing to Wikiart style or using common corruptions. This boost in the AUC is attributed to the style randomization and consistency losses used during NSS training that acts as regularizers and prevents the classifiers from overfitting to specific image styles.

Results in Fig. 3 and Figs. 8, 9, 10 (in the Appendix) show that classifiers trained with NSS, when evaluated with TT-NSS, achieve better accuracy on non-abstained samples for different abstaining rates and in most cases achieve competitive performance with classifiers trained with RSC and SagNet. While in our work we used NSS with ERM, it can be combined with any other DG method such as RSC or SagNet to improve their accuracy on non-abstained samples at different abstaining rates. Moreover, training the classifiers with NSS improves the performance of the confidence-based abstaining mechanism as shown in Tables 8 and 9 (in the Appendix) but even then TT-NSS remains superior in case of severe shifts (such as severity 5 corruptions).

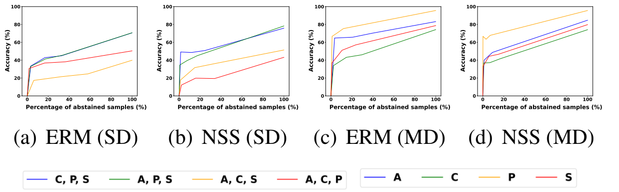
4.3. Predictions on abstained samples

Here we evaluate the effectiveness of TT-NSS in correctly abstaining on samples that could lead to misclassifications. We show this by showing the accuracy of the DG classifier on the test samples that were abstained. Results



(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

Figure 3. Effectiveness of using NSS (with ERM) (solid lines) at producing better risk-averse predictions when evaluated with TT-NSS in comparison to that of other DG methods (dashed lines) in a **single** domain setup. NSS-trained classifiers achieve significantly better accuracy on non-abstained samples compared to classifiers trained with ERM and achieve competitive performance to classifiers trained with RSC and SagNet at different abstaining rates on variants of the **PACS** dataset. (See Fig. 2 for the explanation of setting.)



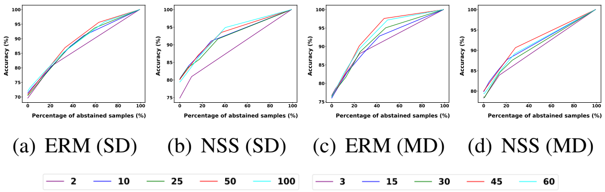
(a) ERM (SD) (b) NSS (SD) (c) ERM (MD) (d) NSS (MD)

Figure 4. Accuracy on samples abstained from a prediction by TT-NSS in single (SD) (a, b) and multiple (MD) (c,d) domain settings on the PACS dataset. (Test domains are denoted in the legend.)

in Fig. 4 show that using a small value of the threshold α where TT-NSS abstains on few samples, the accuracy on abstained samples is significantly lower for classifiers trained with ERM and NSS in both single and multiple source domain settings on the PACS dataset (original style). This is in comparison to the standard accuracy of the classifier (recovered at 100% abstaining rate). The low accuracy on abstained samples suggests that TT-NSS correctly refrains from making predictions on ambiguous samples. Moreover, the accuracy on abstained samples decreases for most test domains for classifiers trained with NSS compared to classifiers trained with ERM, suggesting that NSS improves the ability of TT-NSS to identify risky samples.

4.4. Effect of number of styles

Here we evaluate the effect of using different numbers of re-stylizations of a single test image, n, in TT-NSS using a subsample (see App. B.2) of the PACS dataset (original style). Results in Fig. 5 show that in both single and multi-source domain settings, using a large value of n leads to only a small improvement in the accuracy of non-abstained samples at higher abstaining rates whereas performance at lower abstaining rates remains similar for different values of n. Since using a larger value of n can slow down the inference, we use n as 10 and 15 (5 per domain) in the single and multiple source domain settings. Evaluating a single test sample with TT-NSS using 15 styles increases the inference cost by mere 0.26 seconds on our hardware, showing the potential of TT-NSS at producing risk-averse predictions without sacrificing inference efficiency.



(a) ERM (SD) (b) NSS (SD) (c) ERM (MD) (d) NSS (MD)

Figure 5. The performance of TT-NSS is not significantly affected by the value of n beyond $n = 10$ for single (SD) (a, b) and $n = 15$ in multiple (MD) (c, d) source domain settings. For the SD setting, the classifier is trained on the Cartoon domain and evaluated on the remaining domains in PACS, and for the MD setting, the classifier is evaluated on the Cartoon domain after training on the rest.

5. Discussion and conclusion

Our work proposed and demonstrated the effectiveness of incorporating an abstaining mechanism based on NSS to improve the reliability of a DG classifier's predictions on data from unseen domains. Using advances in neural style transfer, our inference procedure uses the prediction consistency of the classifier on stylized images to predict or abstain on a test sample and requires only black-box access to the DG classifier. Moreover, we proposed a training procedure to improve the reliability of a classifier's prediction at different abstaining rates and demonstrated its effectiveness on various datasets and their variations. We note that while NSS is effective at gauging the reliability of a classifier's prediction on test samples, ascertaining the robustness of this prediction to arbitrary style changes is an important open problem and will be the focus of future works.

6. Acknowledgment

This work was supported by the NSF EPSCoR-Louisiana Materials Design Alliance (LAMDA) program #OIA-1946231 and was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE- AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 23-ERD-030.

References

[1] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019. 1, 3

[2] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4845–4854, 2019. 1

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018. 3

[5] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018. 1

[6] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(8), 2008. 3

[7] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1

[8] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 1

[9] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007. 1

[10] Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019. 1

[11] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020. 1

[12] Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021. 3

[13] Nontawat Charoenphakdee, Zhenghang Cui, Yivan Zhang, and Masashi Sugiyama. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, pages 1507–1517. PMLR, 2021. 3

[14] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10–17, 2018. 3

[15] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. 3

[16] Caroline Choi, Fahim Tajwar, Yoonho Lee, Huaxiu Yao, Ananya Kumar, and Chelsea Finn. Conservative prediction via data-driven confidence minimization. *arXiv preprint arXiv:2306.04974*, 2023. 2

[17] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970. 3

[18] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019. 2, 3, 4

[19] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016. 1, 3

[20] Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407*, 2020. 3

[21] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 3

[22] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013. 2, 5, 13

[23] Marc Fischer, Maximilian Baader, and Martin Vechev. Certified defense to image transformations via randomized smoothing. *Advances in Neural Information Processing Systems*, 33:8404–8417, 2020. 3

[24] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 3

[25] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 3

[26] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3985–3993, 2017. 3

[27] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1

[28] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 1, 3, 5

[29] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019. 1, 6

[30] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[31] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 5

[32] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 1, 6

[33] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 2

[34] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 1, 3

[35] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015, 2020. 1

[36] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 1, 3, 4

[37] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pages 124–140. Springer, 2020. 5, 6, 14

[38] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018. 3

[39] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021. 2, 3

[40] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020. 2, 5

[41] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 527–536. PMLR, 2019. 1

[42] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 3

[43] James M Joyce. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer, 2011. 5

[44] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021. 3

[45] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019. 3

[46] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. ., 2018. 3

[47] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 2, 5, 13

[48] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 535–557, 2021. 3

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 5, 7

[50] Ankur Mallick, Chaitanya Dwivedi, Bhavya Kailkhura, Gauri Joshi, and T Han. Probabilistic neighbourhood component analysis: sample efficient uncertainty estimation in deep learning. *arXiv preprint arXiv:2007.10800*, 2020. 6

[51] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009. 1

[52] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. How robust are randomized smoothing based defenses to data poisoning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13244–13253, 2021. 3

[53] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. Understanding the limits of unsupervised domain adaptation via data poisoning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1

[54] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. Do domain generalization methods generalize well? In *NeurIPS ML Safety Workshop*, 2022. 1

[55] Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2020. 3

[56] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021. 1, 5, 6, 14

[57] Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. On the calibration of multiclass classification with rejection. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[58] K. Nichol. Painter by numbers. `https://www.kaggle.com/competitions/painter-by-numbers`, 2016. 4, 5

[59] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 1, 3

[60] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018. 3

[61] Burr Settles. Active learning literature survey. ., 2009. 1

[62] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1

[63] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Encoding robustness to image style via adversarial feature perturbations. *Advances in Neural Information Processing Systems*, 34:28042–28053, 2021. 3

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[65] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2, 5

[66] Jiachen Sun, Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, Dan Hendrycks, Jihun Hamm, and Z Morley Mao. Certified adversarial defenses meet out-of-distribution corruptions: Benchmarking robustness and simple baselines. *arXiv preprint arXiv:2112.00659*, 2021. 2, 3, 5

[67] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 3

[68] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 3

[69] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Selfnorm and crossnorm for out-of-distribution robustness. ., 2020. 3

[70] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 3

[71] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6924–6932, 2017. 3

[72] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 2, 5, 13

[73] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 3

[74] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[75] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021. 1

[76] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5239–5247, 2017. 3

[77] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 3

[78] Guojun Zhang, Han Zhao, Yaoliang Yu, and Pascal Poupart. Quantifying and improving transferability in domain generalization. *arXiv preprint arXiv:2106.03632*, 2021. 1, 3

[79] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6

[80] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, pages 4944–4953, 2018. 3

[81] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P Xing. Towards principled disentanglement for domain generalization. *arXiv preprint arXiv:2111.13839*, 2021. 3

[82] Jize Zhang, Bhavya Kailkhura, and T Han. Leveraging uncertainty from deep learning for trustworthy materials discovery workflows. *arXiv preprint arXiv:2012.01478*, 2020. 6

[83] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022. 2, 3

[84] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019. 1

[85] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31, 2018. 1, 3

[86] Yuyang Zhao, Zhun Zhong, Zhiming Luo, Gim Hee Lee, and Nicu Sebe. Source-free open compound domain adaptation in semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):7019–7032, 2022. 3

[87] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. *arXiv preprint arXiv:2204.02548*, 2022. 5

[88] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. *arXiv preprint arXiv:2207.04892*, 2022. 3

[89] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 3

Appendix

We present the results of additional experiments in Appendix A followed by details of datasets used in our work along with implementation details of our algorithm and baselines in Appendix B.

A. Additional experiments

Here we present the results omitted in the main paper due to space limitations. In App. A.1, we provide additional empirical results for the comparison between TT-NSS and the confidence-based abstaining mechanism on the VLCS dataset and the multi-source domain setting. In App. A.2, we provide additional empirical results demonstrating the effectiveness of models trained with NSS when evaluated with TT-NSS and the confidence-based abstaining mechanism on different datasets in both single and multi-domain settings.

A.1. Additional results on the comparison of TT-NSS and confidence-based abstaining

Here we present results on the comparison of TT-NSS and confidence-based abstaining using the AUC metric, and present results on the VLCS dataset both in single and multi-domain settings. Our results in Tables 3, 4 and 5, 6 show that similar to the results presented in Fig. 2 in the main paper, the AUC for the accuracy versus the percentage of abstained samples curve is significantly better for TT-NSS compared to confidence-based abstaining in the single domain setting and is competitive on the multi-domain setting. The advantage of TT-NSS becomes clear when evaluated on data from Wikiart and corrupted domains. This advantage of TT-NSS holds regardless of the training method used for training the DG classifier or the dataset used.

In Figs. 6 and 7, we show the full accuracy versus percentage of abstained sample curves for classifiers trained on PACS and VLCS dataset in both the single and multi-domain setting. The results show that the performance of the DG classifier when evaluated with TT-NSS remains better or competitive with the performance of the confidence-based abstaining method for most domains and most of the range of abstaining rates.

A.2. Additional results on the effectiveness of NSS at improving risk-averse predictions

In this section, we present additional empirical results on the effectiveness of training DG models with NSS (combined with ERM) on different datasets and settings. Similar to the results in Sec. 4.2 in the main paper, we observe that models trained with NSS achieve consistently better AUC than models trained with ERM on different variants of PACS, VLCS and OfficeHome datasets as shown in Table 7. The highest improvement is observed when classifiers are evaluated on test sets corrupted with severity 5 corruptions. Similar to Fig. 3, we observe in Fig. 8, 9, 10 that NSS trained models achieve better accuracy on non-abstained samples on most domains compared to the models trained with ERM. Incorporating NSS with ERM makes the performance similar to that of other SOTA DG methods such as RSC and SagNet. Due to the versatility of NSS to be combined with any DG method, training classifiers with RSC and SagNet in conjunction with NSS could lead to further improvement in the accuracy of the classifier trained with these SOTA DG methods on non-abstained samples when evaluated with TT-NSS. Lastly, classifiers trained with NSS also perform better in terms of risk-averse predictions when using the confidence-based abstaining mechanism as shown in Tables 8 and 9. As mentioned in Sec. 4.2, TT-NSS remains superior in the presence of severe shifts such as those induced by adding severity 5 corruptions for all the datasets in both single and multi-domain settings.

B. Dataset and experimental details

All codes are written in Python using Tensorflow/Pytorch and were run on an AMD EPYC 7J13 CPU with 200 GB of RAM and an Nvidia A100 GPU. Implementation and hyperparameters are described below.

B.1. Dataset description

In this work, we use three popular benchmark datasets along with their stylized and corrupted version to evaluate the performance of various methods. For single source domain setting, we use 90% of the data for training and 10% for hyperparameter tuning, and for multi-domain setting, we use 80% of the data for training and 20% for hyperparameter tuning.

PACS [47]: This dataset contains images from four domains Art, Cartoons, Photos, and Sketches. It contains 9991 images belonging to 7 different classes.

VLCS [22]: This dataset contains images from four domains Caltech101, LabelMe, SUN09, PASCAL VOC 2007. It contains 10729 images belonging to 5 different classes.

Office-Home [72]: This dataset contains images from four domains Art, Clipart, Product, and Real. It contains 15588 images belonging to 65 different classes.

B.2. Details of the subsample used for reporting the evaluation results in Sec. 4.4

As mentioned in Sec. 4, we use a subsample of the PACS, VLCS and OfficeHome datasets to present the results of using TT-NSS and confidence based abstaining on corrupted variants of the datasets and for the experiment in Sec. 4.4 with different values of n in TT-NSS. For reporting the results on the corrupted version of the dataset we used 10 images per class from VLCS/PACS and 2 images per class

Table 2. Results on single and multi-domain generalization settings using ResNet50 as the backbone on the PACS dataset using RSC [37] and SagNet [56]. The original work, RSC [37], only reports multi-domain results (presented without *) while SagNet [56], only reported results based on the ResNet-18 backbone in the original paper. We used their official implementation using ResNet-50 as the backbone to obtain results for both single and multi-domain settings (reported with *) (see details in Appendix B.3.1).

DG Setting	Methods	A	C	P	S	Avg.
Single	RSC*	72.55	77.30	47.88	57.54	63.82
	SagNet*	77.45	78.36	52.39	53.96	65.54
Multi	RSC	87.89	82.16	97.92	83.35	87.83
	RSC*	85.79	79.60	95.03	81.52	85.49
	SagNet*	86.00	81.29	97.47	80.72	86.37

from the OfficeHome dataset. We report average result over all 10 corruption types for this experiment.

For the experiment in Sec. 4.4 we used the following subsample. For the single source domain setting, we report the results on a balanced subsample of the dataset containing 50 images from each class and each target domain for PACS. For the multi-domain setting, we use 100 images for each class of the target domain for PACS. For classes with fewer samples, we use all the samples from that class

B.3. Experimental details

B.3.1 Reproducing the baselines

For the RSC [37] method, we independently run the code using the official implementation published by the authors, using different configurations (https://github.com/DeLightCMU/RSC). We trained both multi-domain and single-domain RSC [37] classifiers with the same hyperparameters except for smaller batch size 2 and a learning rate of 0.0001 on one random seed. For the SagNet [56], we reproduce their open-source implementation code with the default configuration on three different random seeds (https://github.com/hyeonseobnam/sagnet). We use the official train and test split of PACS for all three methods. Table 2 shows our reproduced results and the results the authors reported in their papers.

B.3.2 Training classifiers with NSS

To train the classifiers with NSS, we incorporate style augmentation and style consistency losses computed on stylized versions of the source domain images generated through the AdaIN decoder. We additionally incorporate the ERM training loss which minimizes the misclassification on original source domain samples. As mentioned in Sec. 3 other losses used in specific DG algorithms can also be incorporated to improve the quality of risk-averse predictions from classifiers trained with those methods. To compute the style consistency loss we use four different styles for every sample in the batch and use a batch size of 16. These losses are then used to fine-tune the ResNet50 backbone augmented with a fully connected layer used for classification. For the multi-domain setting, the classifier that achieves the highest accuracy on the validation set is used for final evaluation whereas for the single source domain setting, the classifier at the last step is used for final evaluation.

Table 3. Comparison of the area under the accuracy versus percentage of abstained samples curve for TT-NSS and the confidence-based abstaining mechanism in a **single** domain setting on different variations of the **PACS** dataset. The training domain is denoted in the columns.

		A	C	P	S
Alg.	Evaluation	Original Style			
ERM	Confidence	**0.882**	0.875	0.634	0.707
	TT-NSS	0.875	0.878	**0.662**	0.702
RSC	Confidence	**0.892**	0.899	**0.705**	0.779
	TT-NSS	0.858	**0.912**	0.682	**0.794**
SagNet	Confidence	**0.913**	**0.91**	**0.741**	0.758
	TT-NSS	0.889	0.88	0.726	**0.771**
		Wikiart Style			
ERM	Confidence	0.84	0.757	0.609	0.558
	TT-NSS	**0.854**	**0.816**	**0.643**	**0.626**
RSC	Confidence	0.823	0.766	0.63	0.662
	TT-NSS	**0.835**	**0.887**	**0.654**	**0.733**
SagNet	Confidence	0.871	0.8	0.683	0.61
	TT-NSS	0.875	**0.813**	**0.692**	**0.718**
		Corrupted with severity 3			
ERM	Confidence	0.832	0.709	0.613	**0.612**
	TT-NSS	**0.886**	**0.812**	**0.622**	0.545
RSC	Confidence	0.871	0.667	0.673	0.62
	TT-NSS	**0.901**	**0.86**	**0.682**	**0.699**
SagNet	Confidence	0.903	0.78	0.725	0.629
	TT-NSS	0.901	**0.794**	**0.731**	**0.667**
		Corrupted with severity 5			
ERM	Confidence	0.696	0.579	0.418	**0.479**
	TT-NSS	**0.834**	**0.708**	**0.519**	0.468
RSC	Confidence	0.728	0.449	0.564	0.465
	TT-NSS	**0.863**	**0.776**	**0.626**	**0.613**
SagNet	Confidence	0.786	0.576	0.565	0.485
	TT-NSS	**0.855**	**0.686**	**0.666**	**0.59**

Table 4. Comparison of the area under the accuracy versus percentage of abstained samples curve for TT-NSS and the confidence-based abstaining mechanism in a **single** domain setting on different variations of the **VLCS** dataset. The training domain is denoted in the columns.

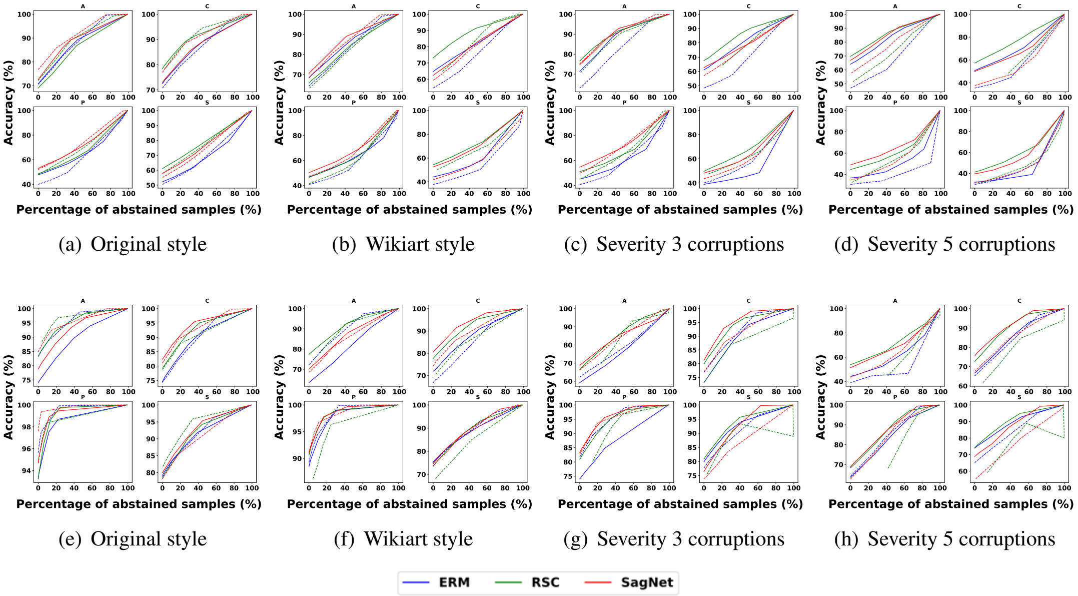
		A	C	P	S
Alg.	Evaluation	Original Style			
ERM	Confidence	**0.653**	0.68	0.806	0.715
	TT-NSS	0.567	**0.724**	**0.851**	**0.751**
		Wikiart Style			
ERM	Confidence	0.426	0.584	0.763	0.679
	TT-NSS	**0.477**	**0.682**	**0.785**	**0.704**
		Corrupted with severity 3			
ERM	Confidence	**0.504**	0.381	**0.734**	0.468
	TT-NSS	0.468	**0.551**	0.689	**0.471**
		Corrupted with severity 5			
ERM	Confidence	**0.433**	0.329	0.563	0.346
	TT-NSS	0.411	**0.439**	0.567	**0.415**

Table 5. Comparison of the area under the accuracy versus percentage of abstained samples curve for TT-NSS and the confidence-based abstaining mechanism in a **multi-**domain setting on different variations of the **PACS** dataset. The domain used for evaluation is denoted in the columns.

		A	C	P	S
Alg.	Evaluation	Original Style			
ERM	Confidence	**0.95**	0.902	0.986	0.915
	TT-NSS	0.893	0.9	0.978	0.911
RSC	Confidence	0.925	0.908	0.978	**0.936**
	TT-NSS	**0.948**	**0.926**	0.983	0.917
SagNet	Confidence	**0.951**	0.932	0.988	0.905
	TT-NSS	0.927	0.939	0.984	**0.925**
		Wikiart Style			
ERM	Confidence	**0.898**	0.85	0.975	0.892
	TT-NSS	0.816	**0.876**	0.97	0.886
RSC	Confidence	0.81	0.842	0.915	0.828
	TT-NSS	**0.911**	**0.916**	**0.976**	**0.891**
SagNet	Confidence	0.858	0.898	0.977	0.886
	TT-NSS	0.869	**0.933**	0.977	0.897
		Corrupted with severity 3			
ERM	Confidence	**0.79**	**0.918**	**0.947**	0.909
	TT-NSS	0.771	0.898	0.878	**0.923**
RSC	Confidence	0.673	0.868	0.802	0.851
	TT-NSS	**0.856**	**0.934**	**0.941**	**0.933**
SagNet	Confidence	0.842	0.913	0.948	0.873
	TT-NSS	0.845	**0.948**	0.953	**0.924**
		Corrupted with severity 5			
ERM	Confidence	0.539	0.85	**0.852**	0.845
	TT-NSS	**0.621**	0.856	0.837	**0.888**
RSC	Confidence	0.405	0.734	0.505	0.673
	TT-NSS	**0.719**	**0.904**	**0.875**	**0.903**
SagNet	Confidence	0.649	0.855	0.845	0.764
	TT-NSS	**0.696**	**0.914**	**0.878**	**0.877**

Table 6. Comparison of the area under the accuracy versus percentage of abstained samples curve for TT-NSS and the confidence-based abstaining mechanism in a **multi-**domain setting on different variations of the **VLCS** dataset. The domain used for evaluation is denoted in the columns.

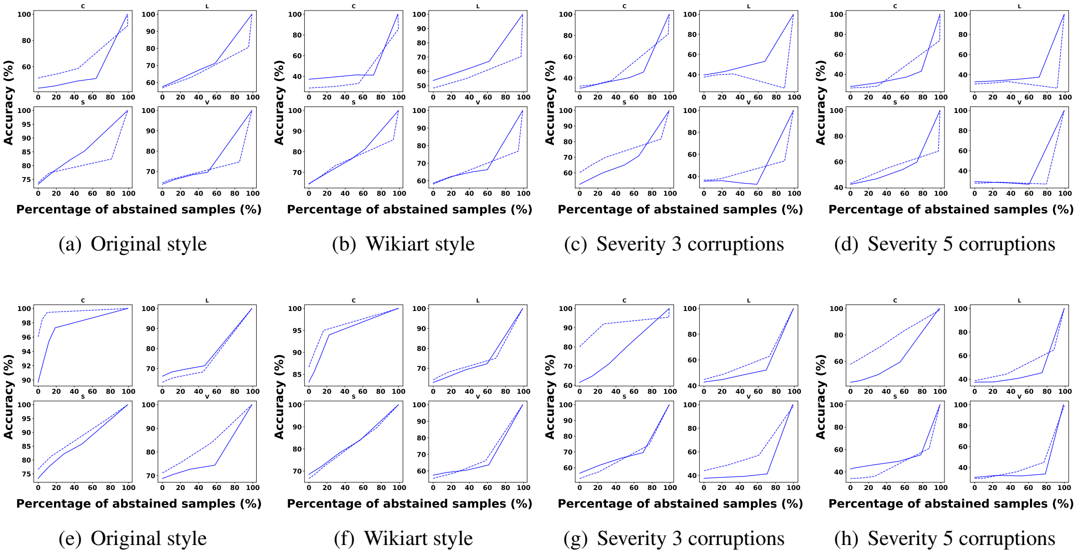
		A	C	P	S
Alg.	Evaluation	Original Style			
ERM	Confidence	0.986	0.752	**0.88**	**0.831**
	TT-NSS	0.968	**0.772**	0.86	0.776
		Wikiart Style			
ERM	Confidence	**0.954**	0.747	0.815	**0.691**
	TT-NSS	0.941	0.744	**0.822**	0.678
		Corrupted with severity 3			
ERM	Confidence	**0.908**	**0.601**	0.678	**0.599**
	TT-NSS	0.785	0.553	**0.692**	0.476
		Corrupted with severity 5			
ERM	Confidence	**0.775**	**0.526**	0.483	**0.427**
	TT-NSS	0.626	0.477	**0.54**	0.388



(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

(e) Original style (f) Wikiart style (g) Severity 3 corruptions (h) Severity 5 corruptions

Figure 6. Comparison of TT-NSS (solid lines) and confidence-based method (dashed lines) in a **single** (top row) and **multi**-source (bottom row) domain setup on classifiers trained with ERM. The graphs show accuracy vs abstained points on different variants of the **PACS** dataset ((a) original, (b) wikiart, (c,d) corrupted), and different source/target domains. In most domains, the accuracy of the TT-NSS (solid line) is similar to or better than the corresponding accuracy of the confidence-based method (dashed line) for most of the range of the percentage of abstained samples. (Note: The source domain from PACS used for training is denoted in the title and the target domain used for evaluation is denoted in the title in the bottom row.)



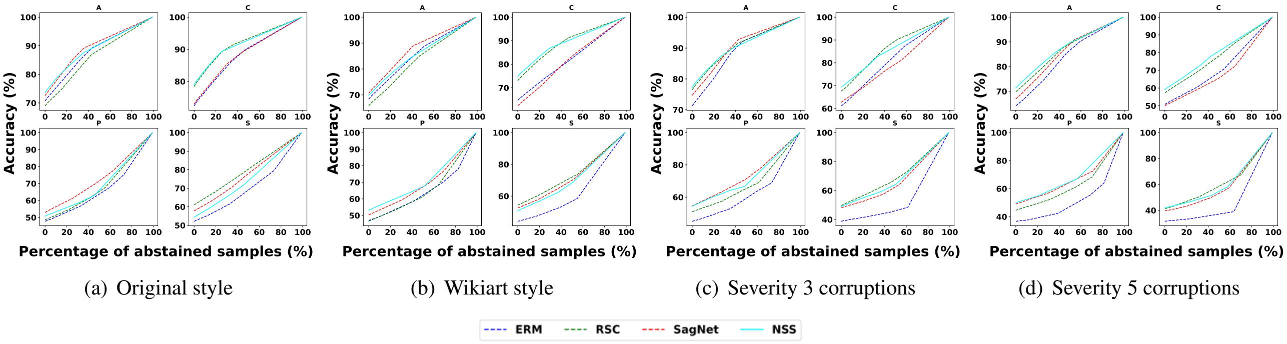
(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

(e) Original style (f) Wikiart style (g) Severity 3 corruptions (h) Severity 5 corruptions

Figure 7. Comparison of TT-NSS (solid lines) and confidence-based method (dashed lines) in a **single** (top row) and **multiple** (bottom row) source domain setup on classifiers trained with ERM. The graphs show accuracy vs abstained points on different variants of the **VLCS** dataset ((a) original, (b) wikiart, (c,d) corrupted), and different source/target domains. In most domains, the accuracy of the TT-NSS (solid line) is similar to or better than the corresponding accuracy of the confidence-based method (dashed line) for most of the range of the percentage of abstained samples. (Note: The source domain from VLCS used for training is denoted in the title in the top row and the target domain used for evaluation is denoted in the title in the bottom row.)

Table 7. Effectiveness of NSS at producing a better AUC score compared to classifiers trained with ERM in a **multiple** source domain setting on PACS, VLCS, and OfficeHome datasets and their variations when evaluated with TT-NSS. (The target domain used for evaluation is denoted in the columns).

	PACS				VLCS				OfficeHome			
Alg.	A	C	P	S	C	S	L	V	A	C	P	R
	Original Style											
ERM	0.893	**0.9**	0.978	0.911	0.968	**0.772**	0.86	0.776	0.683	0.679	0.815	0.83
NSS	**0.95**	0.884	**0.98**	0.914	**0.985**	0.769	0.865	**0.818**	**0.72**	**0.749**	**0.836**	**0.849**
	Wikiart Style											
ERM	0.816	0.876	0.97	0.886	0.941	0.744	0.822	0.678	0.578	0.534	0.692	0.726
NSS	**0.926**	0.869	0.971	**0.909**	**0.98**	**0.766**	**0.85**	**0.775**	**0.667**	**0.713**	**0.798**	**0.825**
	Corrupted with severity 3											
ERM	0.771	0.898	0.878	0.923	0.785	0.553	0.692	0.476	0.5	0.64	0.677	0.715
NSS	**0.889**	**0.933**	**0.943**	**0.933**	**0.959**	**0.605**	**0.706**	**0.632**	**0.587**	**0.697**	**0.738**	**0.812**
	Corrupted with severity 5											
ERM	0.621	0.856	0.837	0.888	0.626	0.477	0.54	0.388	0.387	0.53	0.554	0.59
NSS	**0.792**	0.854	**0.88**	**0.902**	**0.898**	**0.53**	**0.611**	**0.517**	**0.473**	**0.648**	**0.628**	**0.721**



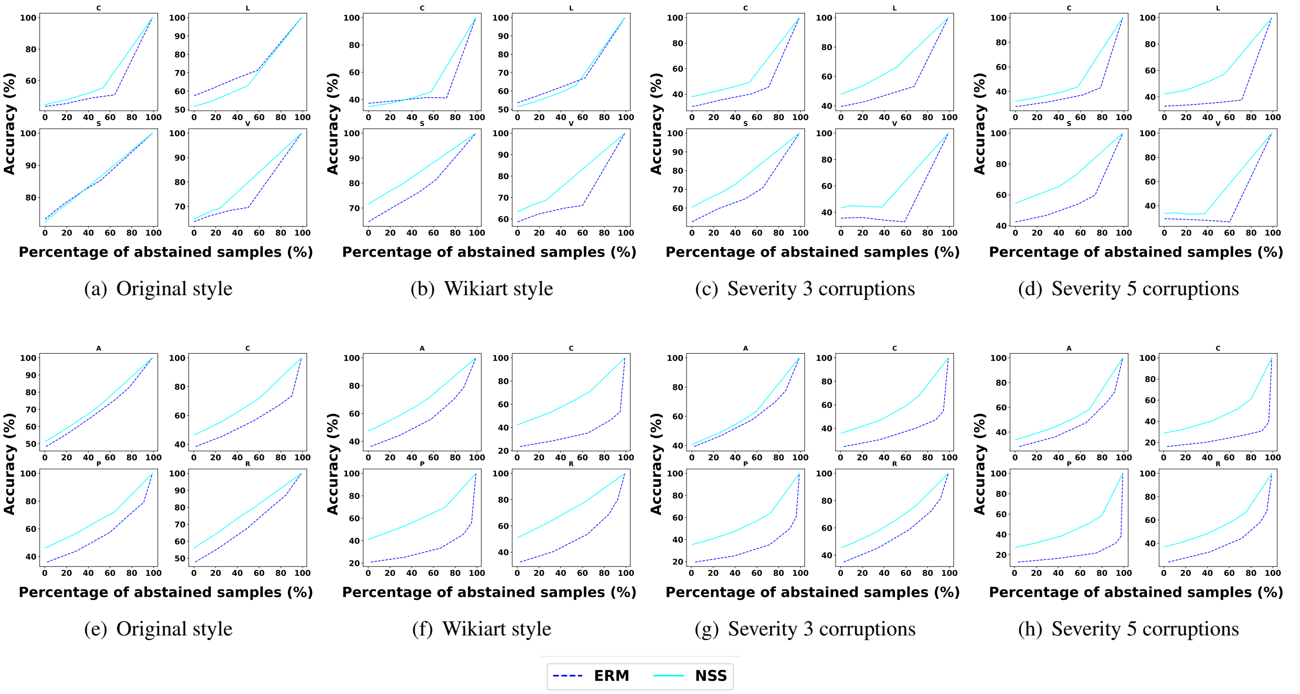
(a) Original style

(b) Wikiart style

(c) Severity 3 corruptions

(d) Severity 5 corruptions

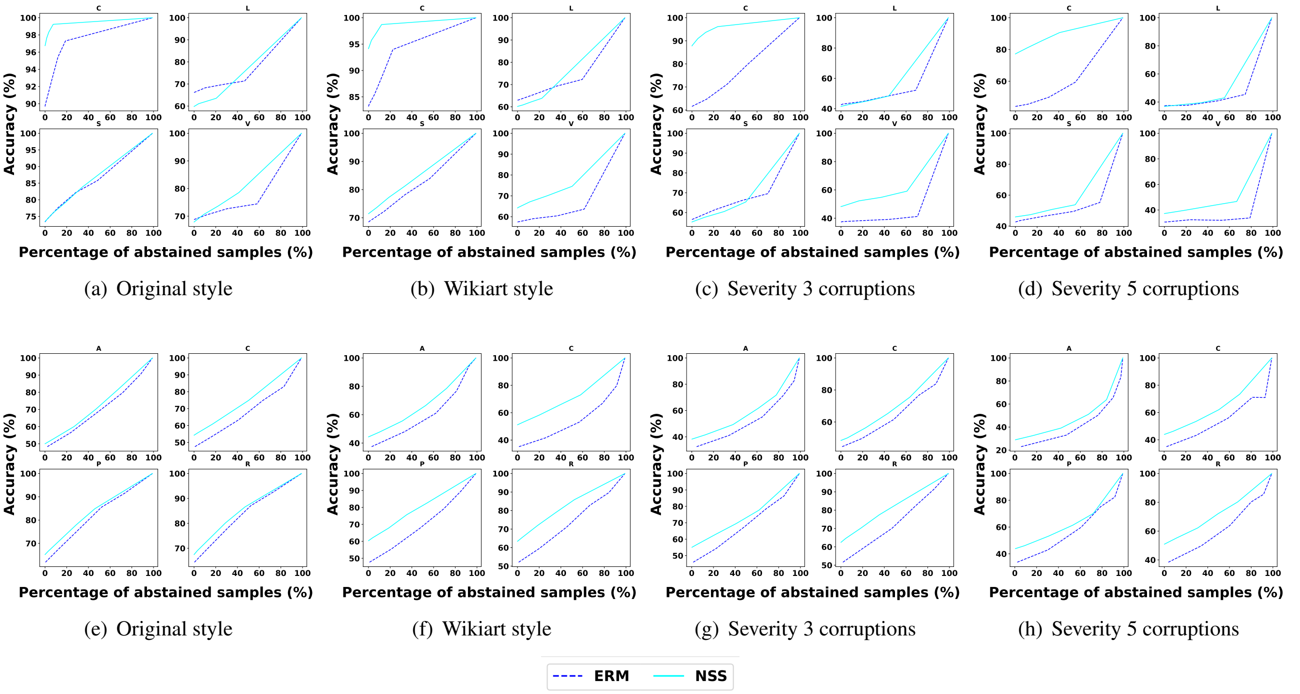
Figure 8. Effectiveness of using NSS (with ERM) (solid lines) at improving the ability of DG classifiers at producing risk averse predictions when evaluated with TT-NSS in comparison to that of other DG methods (dashed lines) in a **multi**-domain setup. NSS-trained classifiers achieve significantly better accuracy on non-abstained samples compared to classifiers trained with ERM and achieve competitive performance to models trained with RSC and SagNet at different abstaining rates on variants of the **PACS** dataset in a multi-source domain setup. (See Fig. 6 for the explanation of settings.)



(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

(e) Original style (f) Wikiart style (g) Severity 3 corruptions (h) Severity 5 corruptions

Figure 9. Effectiveness of using NSS (with ERM as the base DG method) (solid lines) at improving the ability of DG to produce risk-averse predictions when evaluated with TT-NSS making it superior or competitive to classifiers trained with ERM (dashed lines) on variants of the **VLCS** (top row) and **OfficeHome** (bottom row) dataset in a **single** source domain setup. (See Fig. 2 for the explanation of settings.)



(a) Original style (b) Wikiart style (c) Severity 3 corruptions (d) Severity 5 corruptions

(e) Original style (f) Wikiart style (g) Severity 3 corruptions (h) Severity 5 corruptions

Figure 10. Effectiveness of using NSS (with ERM as the base DG method) (solid lines) at improving the ability of DG to produce risk-averse predictions when evaluated with TT-NSS making it superior or competitive to classifiers trained with ERM (dashed lines) on variants of the **VLCS** (top row) and **OfficeHome** (bottom row) dataset in a **multi**-source domain setup. (See Fig. 6 for the explanation of settings.)

Table 8. Effectiveness of NSS at producing a better AUC score compared to classifiers trained with ERM in a **single** source domain setting on PACS, VLCS, and OfficeHome datasets and their variations when evaluated with the confidence-based abstaining mechanism. (The source domain used for training is denoted in the columns).

	PACS				VLCS				OfficeHome			
Alg.	A	C	P	S	C	L	S	V	A	C	P	R
	Original Style											
ERM	0.882	0.875	0.634	**0.707**	0.653	0.68	0.806	0.715	0.743	0.717	0.699	**0.789**
NSS	**0.907**	**0.923**	**0.733**	0.665	**0.671**	0.687	**0.838**	**0.74**	0.739	0.72	0.708	0.778
	Wikiart Style											
ERM	0.84	0.757	0.609	**0.558**	0.426	0.584	0.763	0.679	0.545	0.364	0.334	0.484
NSS	**0.871**	**0.885**	**0.672**	0.526	**0.535**	**0.655**	**0.816**	**0.722**	**0.705**	**0.658**	**0.64**	**0.749**
	Corrupted with severity 3											
ERM	0.832	0.709	0.613	**0.612**	0.504	0.381	0.734	0.468	0.596	0.412	0.411	0.586
NSS	**0.871**	**0.865**	**0.754**	0.549	**0.592**	**0.631**	**0.771**	**0.522**	**0.666**	**0.586**	**0.566**	0.595
	Corrupted with severity 5											
ERM	0.696	0.579	0.418	**0.479**	0.433	0.329	0.563	0.346	0.416	0.243	0.223	0.388
NSS	**0.769**	**0.746**	**0.667**	0.434	0.454	**0.576**	**0.635**	**0.4**	**0.546**	**0.49**	**0.415**	**0.42**

Table 9. Effectiveness of NSS at producing a better AUC score compared to classifiers trained with ERM in a **multiple** source domain setting on PACS, VLCS, and OfficeHome datasets and their variations when evaluated with the confidence-based abstaining mechanism. (The target domain used for evaluation is denoted in the columns).

	PACS				VLCS				OfficeHome			
Alg.	A	C	P	S	C	L	S	V	A	C	P	R
	Original Style											
ERM	0.95	0.902	0.986	0.915	0.986	**0.752**	**0.88**	0.831	0.802	0.721	0.889	**0.905**
NSS	0.955	0.896	0.985	0.922	0.987	0.706	0.86	0.829	0.783	**0.767**	0.876	**0.884**
	Wikiart Style											
ERM	0.898	0.85	0.975	0.892	0.954	**0.747**	0.815	0.691	0.601	0.588	0.726	0.796
NSS	**0.927**	**0.898**	0.982	**0.92**	**0.982**	0.705	**0.833**	**0.781**	**0.707**	**0.747**	**0.829**	**0.838**
	Corrupted with severity 3											
ERM	0.79	0.918	0.947	0.909	0.908	0.601	0.678	0.599	0.529	0.584	0.74	0.717
NSS	**0.887**	0.909	0.955	**0.922**	**0.966**	0.594	**0.735**	**0.627**	**0.647**	**0.735**	**0.775**	**0.808**
	Corrupted with severity 5											
ERM	0.539	0.85	0.852	0.845	0.775	**0.526**	0.483	0.427	0.362	0.475	0.581	0.551
NSS	**0.735**	**0.881**	**0.887**	0.833	**0.91**	0.508	**0.621**	**0.44**	**0.528**	**0.66**	**0.672**	**0.688**