

Appendix

A. Related work

Domain generalization: The goal of domain generalization () 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 (Albuquerque et al., 2019; Zhang et al., 2021a; Ganin et al., 2016; Zhao et al., 2018; Qiao et al., 2020; Gulrajani & Lopez-Paz, 2020). 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 (Arjovsky et al., 2019; Zhang et al., 2021b; Dittadi et al., 2020; Montero et al., 2020). Yet another line of research focuses on diversifying the source domain data to encompass possible variations that may be encountered at test time (Hendrycks et al., 2019; Wang et al., 2021; Kireev et al., 2021; Calian et al., 2021; Sun et al., 2021). Unlike previous works which focus on improving classifier accuracy on unseen domains, we focus on making risk-averse and improving the reliability of classifier predictions on unseen domains.

Certified robustness via randomized smoothing: Many works have demonstrated the failure of SOTA machine learning classifiers on adversarial examples which are crafted by adding imperceptible perturbations to test samples (Szegedy et al., 2013; Chen et al., 2018; Xiao et al., 2018; Chen et al., 2017; Ilyas et al., 2018). In response, many works proposed to provide empirical (Athalye et al., 2018) and provable (Li et al., 2018; Lecuyer et al., 2019; Cohen et al., 2019; Raghunathan et al., 2018; Zhang et al., 2018) robustness to these examples. Among them, Randomized Smoothing (RS) (Li et al., 2018; Lecuyer et al., 2019; Cohen et al., 2019) is one of the popular methods which provides provable robustness to adversarial examples by considering a smoothed version of the original classifier and certifying that no adversarial perturbation exists within a certified radius (in ℓ_2 norm) that can change the prediction of the classifier. RS uses Gaussian noise to produce a smoothed version of the base classifier. For a test sample, it then assigns the label which is 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 (Fischer et al., 2020; Li et al., 2021) where the noise is added to the parameters of the transformations. Our neural style smoothed classifier is in a similar spirit 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 the work of (Gatys et al., 2016), which for the first time demonstrated the effectiveness of using the convolutional layers of CNN for style transfer, several ways have been proposed to achieve better and faster neural style transfer (Gatys et al., 2017; Johnson et al., 2016; Ulyanov et al., 2016; Wang et al., 2017; Ulyanov et al., 2017; Dumoulin et al., 2016). AdaIN (Huang & Belongie, 2017) is a popular approach that allows arbitrary style transfer in real time by changing only the mean and variance of the convolutional feature maps. Other ways of generating stylized images include mixing styles (Zhou et al., 2021), exchanging (Tang et al., 2020; Zhao et al., 2022a) styles, or using adversarial learning (Zhong et al., 2022; Shu et al., 2021).

Test-time adaptation: Recent works have demonstrated the effectiveness of using test-time adaptation for improving generalization to unseen domains, where the classifier is updated on the incoming batch of test samples (Wang et al., 2020; Sun et al., 2020). This approach has also been shown to be effective in the setup (Iwasawa & Matsuo, 2021). Our approach is different from these methods since we do not update the classifier but rather only assume black-box access to it and produce the prediction of the smoothed classifier. Moreover, we use a single test sample, unlike previous methods which assume that the data from various unseen domains arrives in batches at test time.

Classification with abstaining: A learning framework allowing a classifier to abstain on samples has been studied extensively (Chow, 1970; Bartlett & Wegkamp, 2008; Ni et al., 2019; Charoenphakdee et al., 2021; Cortes et al., 2016). 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.

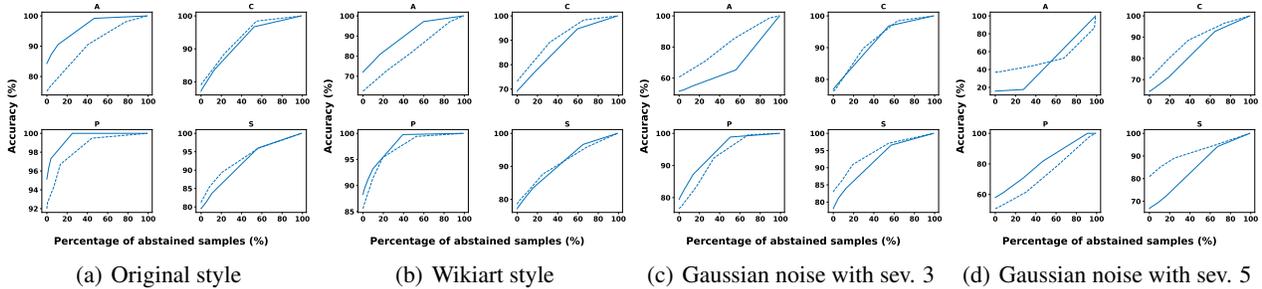


Figure 5. Comparison of TT-NSS (dashed lines) and confidence-based method (solid lines) in a multiple source domain setup. The graphs show accuracy vs abstained points for different datasets ((a) original, (b) wikiart, (c,d) corrupted), and different source/target domains. For most settings, the accuracy of the TT-NSS (dashed line) is higher than the corresponding accuracy of the confidence-based method (solid line) for most of the range of the percentage of abstained points. This demonstrates the superior performance of our style smoothing-based method as opposed to the conventional confidence-based method for producing risk-averse predictions. (Note: The target domain from PACS used for evaluation is denoted in the title.)

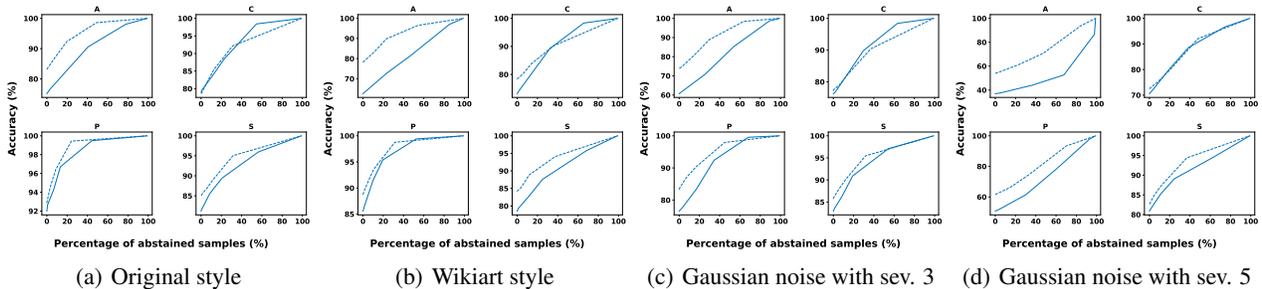


Figure 6. Comparison of NSS training (dashed lines) vs ERM training (solid lines) in the multiple source domain setup. (See Fig. 5 for the explanation of settings.) NSS-trained classifiers evaluated with TT-NSS produce better accuracy on non-abstained samples at different abstaining rates compared to ERM-trained classifiers in the multiple source domain setup on different variants of the PACS dataset.

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 the PACS dataset comprising of 9991 images belonging to 7 categories from four domains Art, Cartoons, Photos, and Sketches along with its 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 multiple source domains setting, we use 80% of the data for training and 20% for hyperparameter tuning.

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

As mentioned in Sec. 3, to speed up the evaluation process when using TT-NSS, we present results on a subsample of the target domain. This approach has been used to report the results in previous works related to randomized smoothing (Cohen et al., 2019; Sun et al., 2021; Zhai et al., 2020; Mehra et al., 2021a). 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 multiple source domains 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. This subsample is used to report the results for the dataset in the original style and the Wikiart style. For reporting results on the corrupted version of the dataset, we create a balanced subsample of roughly one-fifth of the samples chosen for other styles (e.g. we used 10 images per class for each target domain in a single source domain setting for PACS) and report the results by averaging over all ten corruption types.

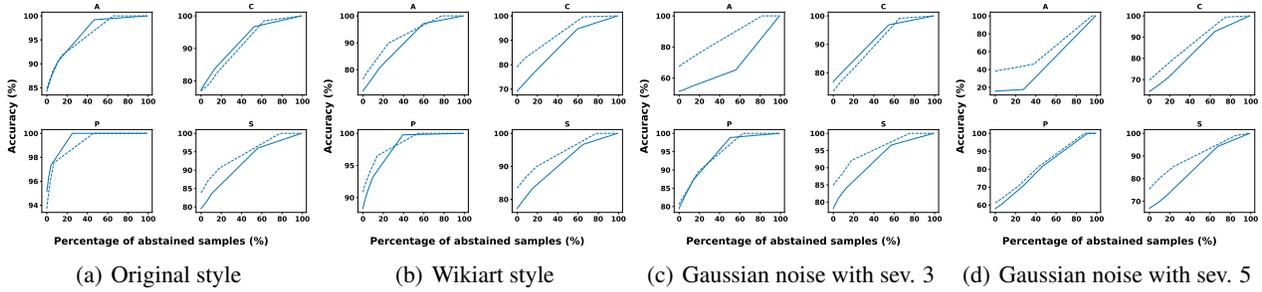


Figure 7. Comparison of NSS training (dashed lines) vs ERM training (solid lines) in multiple source domain setup. (See Fig. 5 for the explanation of settings.) NSS-trained classifiers when evaluated with the confidence-based method also produce better accuracy on non-abstained samples at different abstaining rates compared to ERM-trained classifiers in the multiple source domains setup on different variants of the PACS dataset.

B.3. Experimental details

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. 2 other losses used in specific 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 multiple source domains 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.