Center-aware Adversarial Augmentation for Single Domain Generalization

Tianle Chen Mahsa Baktashmotlagh Zijian Wang
The University of Queensland
{firstname.lastname}@uq.edu.au

Mathieu Salzmann
EPFL
mathieu.salzmann@epfl.ch

Abstract

Domain generalization (DG) aims to learn a model from multiple training (i.e., source) domains that can generalize well to the unseen test (i.e., target) data coming from a different distribution. Single domain generalization (Single-DG) has recently emerged to tackle a more challenging, yet realistic setting, where only one source domain is available at training time. The existing Single-DG approaches typically are based on data augmentation strategies and aim to expand the span of source data by augmenting out-of-domain samples. Generally speaking, they aim to generate hard examples to confuse the classifier. While this may make the classifier robust to small perturbation, the generated samples are typically not diverse enough to mimic a large domain shift, resulting in sub-optimal generalization performance. To alleviate this, we propose a center-aware adversarial augmentation technique that expands the source distribution by altering the source samples so as to push them away from the class centers via a novel angular center loss. We conduct extensive experiments to demonstrate the effectiveness of our approach on several benchmark datasets for Single-DG and show that our method outperforms the state-of-the-art in most cases.

1. Introduction

Most of the existing machine learning algorithms work based on the assumption that the training and test data follow similar distributions. In computer vision, this assumption can be easily violated due to a change in style (e.g., illumination, background) of the images in the training and test data. [18, 19, 5, 4] Such a phenomenon is termed as domain shift and leads to severe performance degradation. To address this problem, domain adaptation methods aim to align the distributions of the labeled source data and the unlabeled target data by learning a discriminative yet domain-invariant representation. [2, 1, 17] Domain generalization (DG) tackles the more challenging scenario where the target data is not available during training by seeking to learn a model that can perform well on the data coming from unseen test domains.

In the standard setting, DG assumes the availability of multiple source domains at training time. In real scenarios, however, collecting and annotating data from different environments is expensive and time-consuming, which hinders the application of DG methods. Therefore, recent research studies the more realistic DG setting where only one source domain is available in the training stage. This is referred to as single domain generalization (Single-DG).

Due to the lack of access to diverse source domains at training time, most of the existing Single-DG methods aim to address unseen domain shifts by generating data from fictitious yet challenging domains. One of the most effective ways to do so is adversarial data augmentation [27, 24, 35]. In essence, it consists of generating hard examples by maximizing the classification error and the entropy of the predictions. While this creates hard samples, such samples do not necessarily accurately mimic a domain shift; they can rather be thought of as adversarial perturbations. For example, in the context of digit recognition, adversarial augmentation may try to confuse the classifier by augmenting a '0' so that it looks like an '8'. While this will make the model robust to such 'hard' examples, it will not reduce its vulnerability to large distribution shifts, such as that occurring between real photographs and sketches. To address this, we propose a Center-aware Adversarial Domain Augmentation (CADA) approach. As illustrated in Figure 1, it generates new, diverse data by modifying the source samples so as to push them away from the class center, thus expanding the source distribution. This is achieved via a novel angular center loss, which calculates the geodesic distance between the original and augmented samples in a latent space. Motivated by the observation that maximizing the margin between classes improves the robustness and generalization power of a model [8, 30], we use our angular center loss not only to expand the data distribution, but also to encourage the classes to be well separated. In other words, to improve generalization, CADA iterates between generating diverse data and enlarging the margin between the different classes in an adversarial manner. We conduct extensive experiments on benchmark object recognition datasets

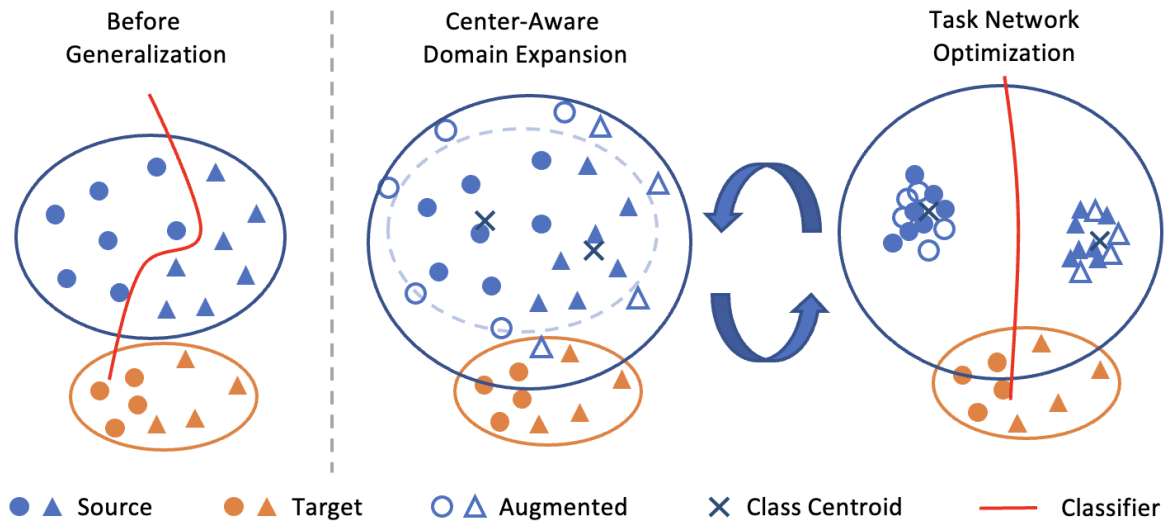


Figure 1: Overall framework of the proposed Center-aware Adversarial Domain Augmentation (CADA). It enhances the generalization power of the task model in an adversarial min-max manner. Previous methods try to generate hard examples to confuse classifier, which may not sufficiently expand the distribution of the source data but rather create samples close to the decision boundary. By contrast, our method pushes the samples away from the class center to expand the source distribution, thus increasing the chances of covering the span of target data. Furthermore, it maximizes the margin between the classes by encouraging the samples to cluster around the class centers.

to demonstrate the effectiveness of our approach, and show that our method outperforms the current state-of-the-arts in most cases.

2. Related Work

Domain generalization (DG) [22] uses multiple training (i.e., source) domains to learn a model that can generalize to the test (i.e., target) data coming from an unseen domain. Single domain generalization (Single-DG) has recently emerged to tackle the more challenging, yet realistic setting, where only one source domain is available at training stage, and the learnt model is tested on unseen target domains. The existing Single-DG approaches typically exploit adversarial data augmentation.

In particular, DG approaches that are based on augmentation and self-supervision have been shown to be successful in the Single-DG setting. Among them, the Representation Self-challenging (RSC) method [10, 28] penalizes the features that generate high gradients in backpropagation, so that all features can get involved in the classification task; JiGen [3] leverages self-supervised learning strategy for data augmentation by splitting the image into tiles, shuffling them, and training the model to perform the main task (i.e., classification), and simultaneously predict the order of the shuffled tiles; [6] introduces a novel normalization layer that can adaptively re-scale the data distribution to improve the model generalization performance across domains. While these methods achieve reasonable performance in the Single-DG case, they were designed for the standard DG scenario.

By contrast, several methods have been proposed specifically to tackle the case of a single source domain. These methods focus on expanding the source data by generating out-of-domain samples, aiming to cover the span of the target data. In particular, ADA [27] is an adaptive data augmentation method that appends adversarial examples to the source data at each iteration; M-ADA [24] follows the MAML-based meta-learning scheme and uses a Wasserstein Auto-Encoder (WAE) to learn a robust model by enlarging the source domain; ME-ADA [35] generates samples which maximize the entropy of the classifier, and exploits meta-learning to progressively augment the training domain with those samples; [32] augments the source data by altering the texture and color of the model's input via a stack of randomly-initialized convolution layers; [29] improves [32] by synthesizing images from more diverse distributions that are complementary to the source one.

In essence, the aforementioned adversarial augmentation methods for Single-DG try to generate hard examples to confuse the classifier or maximise its entropy. This, however, may not expand the distribution of the source data in the way a real domain shift would, as the resulting sample tend to remain close to the decision boundaries. By contrast, our approach more directly expands the source distribution by generating samples that are away from the class center. As will be shown by our experiments, this leads classifiers

that generalize to an unseen target domain.

3. Methodology

In this section, we first define the problem and notation used in this paper, and then provide a detailed discussion of our CADA framework. The main goal of CADA is to expand the source distribution by gradually pushing augmented samples away from the corresponding class centroids. Simultaneously, we aim to a learn latent representation that maximizes the margin between different classes, thus enhancing the robustness and generalizability of the model. To this end, we introduce an angular center loss that measures the geodesic distance between each sample and its corresponding class center in the latent space. CADA then iterates between two tasks in an adversarial manner: the auxiliary task of expanding the source distribution by maximizing the angular center loss, and the main task of creating compact clusters by minimizing the same loss.

3.1. Problem Definition

Let us assume that we have a source domain $D_s = \{(x_1, y_1), ..., (x_n, y_n)\}_{i=1}^n$, with x_i denoting the i-th example samples from distribution P_s, and y_i denoting the label of i-th sample from label space $y \in \{1, ..., k\}$, where k is the number of classes. The target domain D_t is composed of data sampled from distribution P_t, which differs from P_s, but has the same label space as the source domain.

Single-DG is defined as solving the supervised learning problem of training a model on a single source domain so that the model performs equally well on different unseen target domains. Our model, denoted as f, consists of a feature extractor f_e and a classifier f_c, parameterized by θ_e and θ_c, respectively.

3.2. Center-aware Domain Expansion

A natural strategy to mitigate the domain shift in a Single-DG approach is to generate diverse data and incorporate it in the training process. Following this intuition, CADA expands the distribution of the source domain by augmenting and pushing the data away from the centers of their corresponding classes. Specifically, we introduce an angular center loss (AC) to compute the geodesic distance between a sample and its corresponding class centroid in the latent space.

To this end, we use the **angular distance** defined as

$$A(v_1, v_2) = \frac{\arccos D(v_1, v_2)}{\pi}, \tag{1}$$

where v_1 and v_2 are two vectors of the same length and

$$D(v_1, v_2) = \frac{v_1}{\|v_1\|_2} \cdot \frac{v_2}{\|v_2\|_2} = \frac{\sum_{i=1}^n p_i q_i}{\sqrt{\sum_{i=1}^n p_i^2} \sqrt{\sum_{i=1}^n q_i^2}},$$

is the cosine similarity, with p_i and q_i, $i = 1, 2, ..., n$, the elements in vectors v_1 and v_2, respectively.

Angular center loss: Inspired by [30], we initialize the class centers $C = \{c_1, c_2, ..., c_k\} \in \mathbb{R}^{k \times d}$ by randomly sampling k vectors of length d from a Gaussian distribution $N(0, 1)$. We compute the AC loss as the mean of the angular cosine distance between the data and the class centers they belong to. This is expressed as

$$L_{ac}(X, Y, C) = \frac{1}{n} \sum_{i=1}^{n} A(x_i, C[y_i]) \ . \tag{2}$$

During training, the model learns to map the input to feature vectors with minimum angular cosine distance to the corresponding class centers. The centers are iteratively updated to fit the data using the AC loss.

Adversarial data augmentation: The generation process is performed by iteratively updating each input sample and pushing it away from the class centroid, thus aiming to cover the span of the target data. Specifically, we copy the input X as $X^{'}$ and regard the pixels in $X^{'}$ as trainable parameters. During the generation process, we freeze the class centres and the parameters of f_e and f_c. We then maximize the AC loss between $X^{'}$ and C to update the pixels in $X^{'}$ so that they move away from the corresponding class centers. This encourages $X^{'}$ to display novel characteristics compared to X.

We also use X as an anchor and minimize the mean squared error between X and $X^{'}$ to preserve the semantic information in X. This yields a loss to update X defined as

$$\begin{aligned} L_{exp}(X, X^{'}, Y, C) =& L_{mse}(X, X^{'}) - L_{ac}(X^{'}, Y, C) \\ =& ||X - X^{'}||_2^2 - \frac{1}{N} \sum_{i=1}^{N} A(x_i, C[y_i]) \ . \end{aligned} \tag{3}$$

This process is repeated for a few iterations to get the final augmented sample $X_{exp}^{'}$ as

$$X_{exp}^{'} = argmin_{X^{'}} L_{exp}(X, X^{'}, Y, C) \ . \tag{4}$$

3.3. Main Task Optimization

According to [33], a larger class margin improves the out-of-distribution generalization power of a model. Although optimizing the cross entropy loss can learn class-separable features, it does not guarantee the margin to be large[8]. Therefore, we further incorporate our AC loss as a regularization term to the standard cross entropy loss to form compact clusters in the latent space, so that the margin between classes is enlarged. This gives us the final loss to

update the network parameters $\theta = \{\theta_e, \theta_c\}$ and the class centers C as

$$L_{main}(X, Y, C) = L_{ce}(X, Y) + L_{ac}(X, C) \ . \quad (5)$$

3.4. Overall Objective Function

We adopt a two-step iterative training strategy to optimize the augmented samples and the main task. During the center-aware domain expansion step, CADA generates adversarial samples in each class that are far from the corresponding class centers. This is achieved by solving the optimization problem

$$\min_{X'} L_{exp} \ , \quad (6)$$

for all the source samples.

During the main-task optimization step, the objective is to learn discriminative representations with high intra-class compactness to enlarge the margin among classes. This is achieve via th optimization problem

$$\min_{\theta, C} L_{main} \ . \quad (7)$$

Our learning process is summarized in Algorithm 1.

Algorithm 1 CADA

Input: Source domain dataset $D_s = \{(x_i, y_i)\}_{i=1}^n$; Pretrained task model θ; Number of training epochs E; List of augmentation epochs $\mathcal{E}_a$; Number of augmentation iterations T_a
Output: Learnt model parameters θ
Initialize class centres C from standard Gaussian distribution
for e = 1,...,E **do**
  if e in $\mathcal{E}_a$ **then**
    for (X, Y) in D_s **do**
      $X^{'} \longleftarrow X$
      for $t = 1, \ldots, T_a$ **do**
        Compute $L_{exp}(X, X', Y, C)$ by Eq.(3.2)
        $X^{'} \longleftarrow X' - \nabla L_{exp}(X, X', Y, C)$
      end for
      $D_s = D_s \bigcup \{(X^{'}, Y)\}$
    end for
  end if
  for (X, Y) in D_s **do**
    Compute $L_{main}(X, Y, C)$ by Eq.(3.3)
    $\theta \longleftarrow \theta - \nabla L_{main}(X, Y, C)$
  end for
end for

4. Experiments

In this section, we compare our approach with the existing DG and Single-DG baselines on several benchmark single domain generalization datasets. The models are trained with the data coming from only one domain, and evaluated on the remaining domains.

For fair comparison with the existing approaches, in all experiments, we use the same backbone network as employed by the baseline methods. We resize the input to match the different backbones, and follow the same data augmentation process as the state-of-the-art methods.

Evaluation metrics: We compute the mean accuracy on each target domain. We also generate class activation maps (CAM) [36] from the output of the last layer before the classifier to visually show our approach's ability to localize the discriminative regions of interest in the test images. To indicate the models' generalization performance, we display the t-SNE[25] of the extractor's output. We also use t-SNE to show the difference between the original sample and the updated sample to compare existing adversarial sample generalization methods with ours.

4.1. Datasets

Digits is a benchmark single domain generalization dataset consists of MNIST[14], SVHN[23], MNIST-M[7], USPS[11], and SYN[7]. Each domain contains 10 classes of digits from 0 to 9. We train the model on the MNIST as source domain and test it on the other domains. The images are resized to 32×32 pixels, with grayscale images being channel-wise duplicated in MNIST and USPS.

MNIST-C[21] is a benchmark Single-DG dataset [32]. Following the same setting of [32], we use MNIST as the source domain and 16 corruption types in MNIST-C as target domains. All images are resized to 32×32 with grayscale images being channel-wise duplicated.

CIFAR-10-C [9] is commonly-used as a benchmark dataset for Single-DG. The source data comes from the CIFAR-10[13] training set [13], and the test data is from **CIFAR-10-C** [9], consisting of 19 corruption types with five severity levels (5 indicating the most corrupted one), applied to the CIFAR-10 test set [13]. As existing Single-DG baselines, for our target domain, we select 15 corruption types with severity level 5, belonging to the four broad categories: noise, blur, weather and digital.

CIFAR-100-C is composed of the same samples and corruption types as in CIFAR-10-C, categorised in 100 classes. Following the same setting as in [35], we use CIFAR-100 [13] as the source data for our target domain, we select 15 corruption types in all severity level.

PACS [16] is a DG benchmark dataset that contains 9,991 samples from seven classes in four domains (i.e., Art painting, Cartoon, Photo, and Sketch). We use one of these four

domains as the source data and test the model on another domain, forming four different training and test domains pairs. All the images are resized to 224×224 and processed with multiple data augmentation methods. They include crops of random sizes and aspect ratios, random horizontal flips, random colour jitter (0.4 brightness, 0.4 contrast, 0.4 saturation and 0.4 hue), image grayscaling (0.1 probability), and normalization using the ImageNet channel means and standard deviations.

4.2. Implementation Details

Below, we provide the experimental settings and implementation details for each benchmark dataset.

Digits. We use the ConvNet architecture of [15] (conv-pool-conv-pool-fc-fc-softmax) with ReLU following each convolution. We apply the random convolutions of [32] to augment the training data. The ADAM optimizer is used to train the model with a learning rate set to 1e-4, and a batch size of 32. The learning rate for updating the centers is 0.5, using the SGD optimizer. We generate domain adversarial samples every iteration with the SGD optimizer and a learning rate of 1. The generation process takes 10 iterations to finish. **MNIST-C.** We use the same data augmentation and training setting as for Digits. The model is trained on MNIST only and tested on MNIST-C.

CIFAR-10-C. We use a WideResNet [34] with 16 layers and a depth of 4 as backbone. Inspired by [6], the batch normalization layers are replaced with instance normalization layers in each residual block of the backbone. We train the backbone and classifier using the ADAM optimizer. The learning rate is set to 1e-3 and is adjusted between 0 and 1e-3 with a cosine annealing scheduler after each epoch. The centers are updated with the SGD optimizer, with a learning rate of 0.5. We generate domain adversarial samples in each iteration every 5 epochs, and the learning rate is set to 1.0.

CIFAR-100-C. We use WideResNet[34] with 40 layers and a depth of 2 as the backbone. The batch normalization layers in each residual block of the backbone are replaced with instance normalization layer. SGD with learning rate of 1e-2 is used to train the extractor and classifier. A cosine annealing scheduler is also applied to adjust the learning rate between 0 and the initial learning rate. The centers are updated with the SGD optimizer with a learning rate of 0.5. The domain adversarial sample generation is conducted in each iteration every 5 epochs with a learning rate set to 0.5.

PACS. Following the existing baselines, we use a pre-trained ResNet18 as backbone. We use ADAM as the optimizer with an initial learning rate of 1e-5, betas of 0.9 and 0.999, and a cosine annealing schedule to adjust the learning rate in range 0 to the initial learning rate periodically during training. We update the centers with an SGD optimizer, with a learning rate of 5e-3. We generate domain adversarial samples in each iteration every 5 epochs, with the learning rate set to 0.5. The training batch size is 16. The model is trained on each of the domains separately and evaluated and averaged on all the others.

Table 1: Classification accuracy (%) on Digits. The bold numbers indicate the best performance in each column.

	SVHN	MNIST-M	SYN	USPS	Avg.
ERM[12]	27.83	52.72	39.65	76.94	49.29
CCSA[20]	25.89	49.29	37.31	83.72	49.05
d-SNE[31]	26.22	50.98	37.83	93.16	52.05
JiGen[3]	33.80	57.80	43.79	77.15	53.14
ADA[27]	35.51	60.41	45.32	77.26	54.62
M-ADA[24]	42.55	67.94	48.95	78.53	59.49
ME-ADA[35]	42.56	63.27	50.39	81.04	59.32
RSDA[26]	47.4	81.5	62.0	83.1	68.5
L2D[29]	62.86	87.30	63.72	83.97	74.46
RSDA+ASR[6]	52.8	80.8	64.5	82.4	70.1
RC[32]	62.07	**87.89**	63.90	84.39	74.56
Ours	**67.27**	78.66	**79.34**	**96.96**	**80.56**

4.3. Experimental Results

Results on Digits: Table 1 shows the results on the Digits benchmark dataset. We compare our method with the following baselines: (1) ERM, the baseline using the cross-entropy loss for training; (2) CCSA[20], improving generalization via latent features regularization; (3) d-SNE[31], using stochastic neighborhood embedding techniques and a modified-Hausdorff distance for both domain adaptation and domain generalization; (4) JiGen[3], improving generalization by self-supervision and teaching the model to predict the tile order of the partitioned image as an auxiliary task; (5) ADA[27], using an adaptive data augmentation method by appending adversarial examples to the source data at each iteration.; (6) M-ADA[24], improving ADA with Wasserstein auto-encoder and meta-learning schema; (7) ME-ADA[35], improving ADA by generating adversarial examples that can maximize the prediction entropy as an extra constraint in the loss function; (8) RSDA[26], designing a data augmentation method based on image transformations that the current model is the most vulnerable to; (9) L2D[29], augmenting the data with modified random convolution and adjusting the mutual information between sample pairs based on the class information; and (10) ASR[6], a novel normalization method using adaptive batch normalization to fit data statistics from different domains.

Our method outperforms both the baselines and the SOTA methods on average. In particular, in comparison to the existing adversarial sample generalization methods ADA, M-ADA, and ME-ADA, our method improves the performance by a large margin on all domains. Moreover,

Table 2: Single domain generalization accuracy (%). The model is trained on MNIST and tested on the 16 corruption types in MNIST-C.

ERM[12]	ADA[27]	ME-ADA[35]	RC[32]	Ours
88.2	92.58	**94.53**	91.62	93.33

Table 3: Single domain generalization accuracy (%). The models are trained on CIFAR-10 and tested on CIFAR-10-C, with corruption severity level of 5. We report the average accuracy for the 4 broad categories of corruption: Weather, Blur, Noise, and Digital. The best performances are highlighted in bold.

	Weather	Blur	Noise	Digits	Avg.
ERM[12]	67.28	56.73	30.02	62.30	54.08
CCSA[20]	67.66	57.81	28.73	61.96	54.04
d-SNE[31]	67.90	56.59	33.97	61.83	55.07
RSC[10]	63.96	61.41	40.48	58.31	58.76
ADA[27]	72.67	67.04	39.97	66.62	61.58
M-ADA[24]	75.54	63.76	54.21	65.10	64.65
ME-ADA[35]	74.44	71.37	66.47	70.83	70.77
L2D[29]	75.98	69.16	73.29	72.02	72.61
RC+BN[32]	74.39	79.79	76.04	75.0	76.37
RC+IN[32]	78.76	80.75	81.95	82.85	81.29
ERM+ASR[6]	-	-	-	-	72.9
ADA+ASR[6]	-	-	-	-	78.4
Ours	**79.93**	**83.77**	**84.84**	**84.84**	**83.57**

our method performs better than random convolution based methods, L2D and RC, on the SVHN, SYN, and USPS target domains.

Results on MNIST-C: We compare our method with the ERM baseline as well as the SOTA Single-DG methods, including ADA [27], ME-ADA [35], and random convolution[32]. The results in Table 2 show that our method outperforms the ERM baseline as well as the SOTA methods on this dataset.

Results on CIFAR-10-C: In Table 3, we report the results on the 15 corruption types of Level 5 severity on CIFAR-10-C dataset. The table shows the average accuracy for the 4 broad categories of corruption: Weather, Blur, Noise, and Digital and the average accuracy over all corruptions. As can be seen from the results, our method outperforms existing methods, with large margin in the blur, noise, and digits corruption categories. This confirms our model's robustness against perturbations added to the input. Our method also outperforms the SOTA ASR method by large 5% margin.

Results on CIFAR-100-C: In Table 4, we report the average accuracy on 4 main corruption categories and the overall accuracy on all 15 corruptions on CIFAR-10-C dataset.

Table 4: Single domain generalization accuracy (%) on CIFAR-100. Th best performances are highlighted in bold. The models are trained on CIFAR-100 and tested on CIFAR-100-C, with corruption severity levels of 1 to 5. We report the average accuracy for the 4 broad categories of corruption: Weather, Blur, Noise, and Digital.

	Weather	Blur	Noise	Digits	Avg.
ERM[12]	24.67	45.75	55.0	56.0	46.78
ADA[27]	33.0	51.75	55.33	58.2	50.97
ME-ADA[35]	52.67	53.0	52.33	56.2	53.93
RC[32]	56.99	57.17	57.88	57.48	57.38
Ours	**59.81**	**59.28**	**60.19**	**59.66**	**59.96**

Table 5: Single domain generalization accuracies on PACS. The models are trained on one domain and tested on the other domains.

	Artpaint	Cartoon	Sketch	Photo	Avg.
ERM[12]	70.9	76.5	53.1	42.2	60.7
RSC[10]	73.4	75.9	56.2	41.6	61.8
ADA[27]	71.56	76.84	52.36	43.66	61.11
ME-ADA[35]	71.52	76.83	46.22	46.32	60.22
RC[32]	73.68	74.88	55.42	46.77	62.69
RSC+ASR[6]	**76.7**	**79.3**	**61.6**	54.6	68.1
Ours	76.33	79.08	61.59	**56.65**	**68.41**

Our approach outperforms the SOTA methods ADA [27], ME-ADA [35], and RC [32] on all corruption types.

Results on PACS:. In Table 5, we demonstrate our results on PACS dataset, where we use one domain for training and the rest three for testing. The reported numbers are the average accuracy across the test domains. Our method achieves very similar results to the SOTA on all target domains, with slightly better results on the Photo domain.

4.3.1 Ablation Study

Effect of the Angular Center Loss: As discussed in the method section, to compute the distance of the samples from the class centers, instead of using the Euclidean distance as in [30], we propose to use an angular distance. In Table 6, we validate this choice by comparing the results obtained by using the center loss versus the angular center loss on 3 datasets, Digits, CIFAR-10-C, and CIFAR-100-C. We use the same training settings as described in the implementation details. The results show that the angular center loss outperforms the center loss by 1-2% on Digits and CIFAR-10-C, and by 5% on CIFAR-100-C, evidencing that the margin of improvement increases as the number of

Table 6: Comparison between center loss and our angular center loss.

	Digits	CIFAR-10-C	CIFAR-100-C
Center Loss[30]	78.22	83.07	56.92
AC Loss	**80.56**	**84.28**	**62.15**

Table 7: Results on CIFAR-10-C with augmenting the training data by sampling around the decision boundary with different variances. The backbone is our modified WideResNet(16-4) described in the implementation details.

s	0.1	0.25	0.5	1.	5.	10.	Ours
	79.74	80.3	80.17	80.16	79.33	77.81	**80.56**

classes in the target domain grows.

4.3.2 Visualization

Class Activation Maps: Figure 2 shows the class activation maps for our method and the state-of-the-art adversarial generation-based method ME-ADA[35]. Both methods are trained on CIFAR-10 dataset and tested on 15 corrupted versions of the same input. The images show that our method focuses on smaller regions than ME-ADA, depicting the relevant semantic information.

t-SNE visualizations of the extracted features from the test dataset of CIFAR-10-C are shown in Figure 3. The test dataset is composed of 1000x15 data points randomly sampled from each of 15 corruptions. The results show that our approach achieved more compact clusters with larger margins between the classes compared to ME-ADA.

Analysis of the Generated Samples: We use t-SNE [25] to visually analyze the distributions of the generated samples in Figure 4(a)-(c). The light blue points show the distributions of two classes. We use ME-ADA and our method to update all the samples from the same class. The updated samples are shown in orange. Figure 4(a) shows that some of the generated samples with ME-ADA shift dramatically and cluster together into other surrounding classes (see the small orange clusters around the blue points). We regard ME-ADA as too aggressive as it may not preserve the discriminative information of the samples during the update stage. By contrast, our method tends to relocate the samples at the boundary between the classes, and can thus be considered as a better way to expand the knowledge space progressively.

Decision Boundary Sampling: We compare our approach to manual augmentation of the training space by generating samples at the boundary between classes, as shown in Figure 4-c. We use the same classes as in the previous visualization and compute the decision boundary by taking the classifier's element-wise means of class-relative weights. Then, we sample features from $N(0, 1)$ and transform them with the computed means and different variances. Table7 shows that the effect of different variances is inconsistent and performs worse than our proposed method.

5. Conclusion

We have introduced an approach to improve a model's generalization and robustness by augmenting the training data with class-aware adversarial samples. We have hypothesized that the generalizability of a model can be related to its robustness to perturbed samples located close to the decision boundary of the classes yet far from the class centers. Motivated by this, in contrast to existing domain adversarial augmentation methods that generate adversarial samples by maximizing the softmax classification error, we have proposed an angular center loss to help generating adversarial samples close to the corresponding class boundary. We consider this as a more direct way of expanding the source data distribution with diverse samples. The adversarial data augmentation process is performed simultaneously with class-margin maximization to also encourage intra-class compactness. We have conducted extensive experiments to demonstrate the effectiveness of our approach on several benchmark datasets.

References

[1] Mahsa Baktashmotlagh, Mehrtash Harandi, and Mathieu Salzmann. Distribution-matching embedding for visual domain adaptation. *Journal of Machine Learning Research*, 17:Article–number, 2016.

[2] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE international conference on computer vision*, pages 769–776, 2013.

[3] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Zhi Chen, Jingjing Li, Yadan Luo, Zi Huang, and Yang Yang. Canzsl: Cycle-consistent adversarial networks for zero-shot learning from natural language. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 874–883, 2020.

[5] Zhi Chen, Yadan Luo, Ruihong Qiu, Sen Wang, Zi Huang, Jingjing Li, and Zheng Zhang. Semantics disentangling for generalized zero-shot learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[6] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217, 2021.

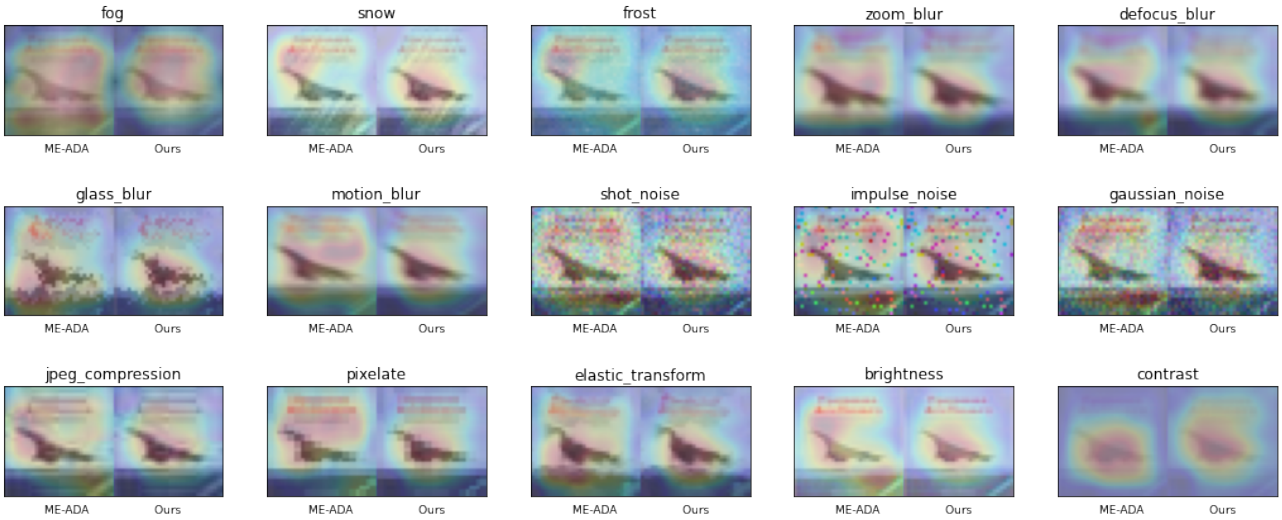
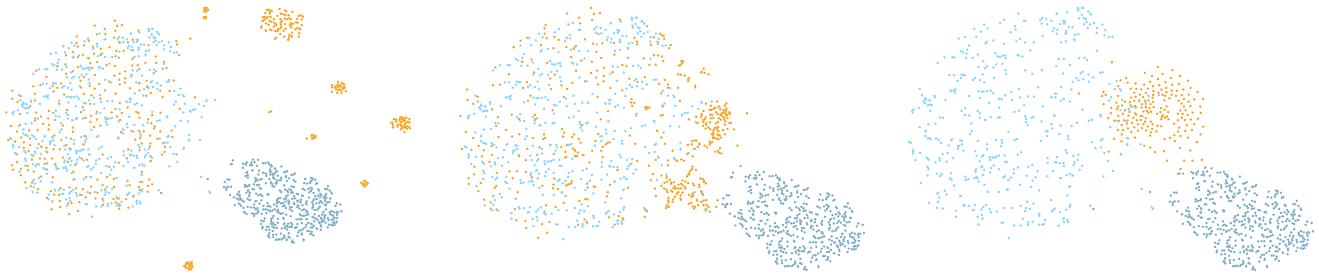


Figure 2: Comparison between the class activation maps of our method and of ME-ADA. On all 15 corrupted versions of the input, our method focuses on a smaller region than ME-ADA, depicting the relevant semantic information.

(a) ME-ADA (b) Ours

Figure 3: t-SNE of extracted features from the test dataset of CIFAR-10-C. (a) ME-ADA; (b) Our method.



(a) ME-ADA (b) Ours (c) Boundary Sampling

Figure 4: t-SNE graphs representing the same two class clusters (light blue and dark blue). (a)-(b) Comparison of ME-ADA and our method. We sample some data points from the left cluster (light blue) and use the corresponding methods to update them. The orange points show the distribution of the updated samples. (c) Data points (orange) randomly sampled around the decision boundary between the two classes.

[7] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on machine learning*, pages 399–406, 2010.

[8] Yiwen Guo and Changshui Zhang. Recent advances in large margin learning. *CoRR*, abs/2103.13598, 2021.

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

[10] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. *European Conference on Computer Vision*, 2020.

[11] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[12] Vladimir Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems. *Lecture Notes in Mathematics*, 2011.

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[14] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[16] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, 2017.

[17] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3071–3085, 2018.

[18] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[19] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning*, pages 6468–6478. PMLR, 2020.

[20] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017.

[21] Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

[22] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, 2013.

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[24] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[25] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

[27] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

[28] Haohan Wang, Songwei Ge, Eric P Xing, and Zachary C Lipton. Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549*, 2019.

[29] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021.

[30] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[31] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. *arXiv preprint arXiv:2007.13003*, 2020.

[33] Yaoqing Yang, Rajiv Khanna, Yaodong Yu, Amir Gholami, Kurt Keutzer, Joseph E. Gonzalez, Kannan Ramchandran, and Michael W. Mahoney. Boundary thickness and robustness in learning models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020.

[34] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[35] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Neural Information Processing Systems*, 2020.

[36] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.