Domain Generalization by Rejecting Extreme Augmentations

Masih Aminbeidokhti, Fidel A. Guerrero Peña, Heitor Rapela Medeiros
Thomas Dubail, Eric Granger, Marco Pedersoli
LIVIA, Dept. of Systems Engineering, ETS Montreal, Canada
{masih.aminbeidokhti.1, fidel-alejandro.guerrero-pena}@ens.etsmtl.ca
{heitor.rapela-medeiros.1, thomas.dubail.1}@ens.etsmtl.ca
{eric.granger, marco.pedersoli}@etsmtl.ca

Abstract

Data augmentation is one of the most effective techniques for regularizing deep learning models and improving recognition performance in various tasks and domains. However, this holds for standard in-domain settings, in which the training and test data follow the same distribution. For the out-of-domain case, where the test data follow a different and unknown distribution, the best recipe for data augmentation is unclear. In this paper, we show that for out-of-domain and domain generalization settings, data augmentation can provide a conspicuous and robust improvement in performance. To do that, we propose a simple training procedure: (i) use uniform sampling on standard data augmentation transformations; (ii) increase the strength transformations to account for the higher data variance expected when working out-of-domain, and (iii) devise a new reward function to reject extreme transformations that can harm the training. With this procedure, our data augmentation scheme achieves a level of accuracy comparable to or better than state-of-the-art methods on benchmark domain generalization datasets. Code: https://github.com/Masseeh/DCAug
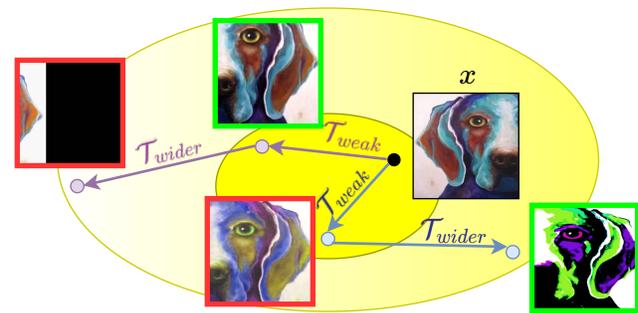
Figure 1. A conceptual illustration of our method. The inner and outer circles represent the space of weak (safe) and wider (possibly harmful) augmentations, respectively. Our method can automatically select for each combination of data samples and augmentation a wider transformation (when safe) or reject it when unsafe. This is achieved with the help of a reward function (represented as the yellow color gradient) that compares the diversity and the consistency of an augmented sample (see Section 4 for more details). In the illustration, given an image x, we present two possible augmentation paths. For the blue path, the wide augmentation has a high diversity and high consistency, and therefore, it is selected (green box). For the purple path, although the wide augmentation has high diversity, it also has low consistency. Therefore the transformation is rejected (red box), and the weak transformation is used instead as augmentation.

1. Introduction

The main assumption of commonly used deep learning methods is that all examples used for training and testing models are independently and identically sampled from the same distribution [44]. In practice, such an assumption does not always hold, and this can limit the applicability of the learned models in real-world scenarios [22].

To tackle this problem, domain generalization (DG) [5] aims to predict data distributions different from those seen during training. In particular, we assume access to multiple datasets during training, each containing examples about the same task but collected under a different domain or environment. One effective approach for DG is to increase the diversity of the training data [50]. Data augmentation, which is a widely used approach for generating additional training data [24, 35, 43], is especially beneficial since it can help to approximate the true distribution of the dataset. However, choosing augmentations often depends on the underlying dataset. While learning the right data augmentation for in-domain setting (same distribution between training and test) has been explored in research [8, 9, 33, 42], there is currently little research on good augmentations for domain generalization and how to leverage domain information to make the augmentations more effective.

In this work, we investigate those questions. First, we show that data augmentation is also helpful for domain generalization, but to cover the different training domains and hopefully the target domain the proposed transformations need to be stronger than for in-domain tasks. However, too strong transformations could harm the learning process (see Figure 1). To fully exploit more robust transformations without harming the learning, we select diverse and challenging samples that provide helpful training information without losing the sample's original semantics. We introduce a reward function consisting of diversity and semantic consistency components and use it to select for each sample the best augmentation between a weak but safe and a strong but diverse augmentation. Thus, the proposed algorithm should be able to select which augmentation is better for each sample for training a model that can generalize to unknown data distributions.

The main contributions of this paper are as follows:

(1) We show that while commonly used augmentation-based techniques for in-domain settings are quite powerful for DG, we can increase the performance further by expanding the range of transformations. Consequently, we achieve superior results compared to most approaches relying on domain-invariant representation.

(2) With the new expanded range, it is easier to produce harmful transformations, therefore, we introduce a data augmentation schema that selects the optimal augmentation strategy between a weak yet safe and a diverse yet strong augmentation technique.

(3) Experimental on common benchmark datasets show the benefits of our proposed method achieving an accuracy that is better than state-of-the-art methods for DG.

2. Related Works

Data Augmentation: There has been extensive research on data augmentation for computer vision tasks. Horizontal flips and random cropping or translations of images are commonly used for natural image datasets such as CIFAR-10 [23], and ImageNet [38]. At the same time, elastic distortions and scalings are more common on MNIST dataset [49]. While data augmentation usually improves model generalization, if too strong, it might sometimes hurt performance or induce unexpected biases. Thus, one must manually find effective augmentation policies based on domain knowledge and model validation. To alleviate this issue, researchers propose various methods to automatically search efficient augmentation strategies for model in-domain generalization [8, 16, 18, 29, 53, 55]. AutoAugment (AA) [8] is the pioneering work on automating the search for the ideal augmentation policy, which uses Reinforcement Learning (RL) to search for an optimal augmentation policy for a given task. Unfortunately, this search process requires extensive computing power to several thou-

sands of GPU hours. Many subsequent works adopt AutoAugment search space for their policy search [50]. In particular, [28, 32] propose methods to shorten the duration of the policy search for data augmentation while maintaining similar performance. Alternatively, other works resort to different guided search techniques to accelerate the search. Lim *et al*. [29] uses a Bayesian optimization approach, [16] uses an online search during the training of the final model, and [18] employs an evolutionary algorithm to search for the optimal augmentation policy also in an online fashion. Adversarial AutoAugment (Adv. AA) [53] is another slightly cheaper method that uses multiple workers and learns the augmentation policy that leads to hard samples measured by target loss during training. However, all of these sophisticated approaches are comparable to RandAugment (RA), which uses the augmentation search space introduced in [9], but with a uniform sampling policy in which only the global magnitude of the transformations and the number of applied transformations are learned on a validation set. TrivialAugment (TA) [33] and UniformAugment [30] further push the RA method to the extreme and propose to use a truly search-free approach for data augmentations selection, yet achieving test set accuracies that are on-par or better than the more complex techniques previously discussed. However, all the mentioned methods use the search space of AutoAugment, which is already designed to not excessively distort the input image. To control the space of data augmentation, Gong *et al*. [13] regularize augmentation models based on prior knowledge while Wei *et al*. [46] use knowledge distillation to mitigate the noise introduced by aggressive AA data augmentation policies. Suzuki [42] proposes an online data augmentation optimization method called TeachAugment that introduces a teacher model into the adversarial data augmentation and makes it more informative without the need for careful parameter tuning. However, all the mentioned methods are designed for standard in-domain settings and do not consider the generalization problem for unknown domains as in domain generalization problems.

Domain Generalization: Learning domain-invariant features from source domains is one of the most popular methods in domain generalization. These methods aim at learning high-level features that make domains statistically indistinguishable (domain-invariant). Ganin *et al*. [12] propose Domain Adversarial Neural Networks (DANN), which uses GAN, to enforce that the features cannot predict the domain. Albuquerque *et al*. [1] build on top of DANN by considering one-versus-all adversaries that try to indicate to which training domain each example belongs. Later work, consider several ways to enforce invariance, such as minimizing the maximum mean discrepancy (MMD) [26], enforcing class-conditional distributions across domains [27], and matching the feature covariance (second-order statis-

tics) across training domains at some level of representation [41]. Although popular, enforcing invariance is challenging and often too restrictive. As a result, Arjovsky *et al.* [2] propose to enforce the optimal classifier for different domains. GroupDRO [39] proposes to minimize the worst-case training loss by putting more mass on samples from the more challenging domains at train time. Bui *et al.* [6] use meta-learning and adversarial training in tandem to disentangle features in the latent space while jointly learning both domain-invariant and domain-specific features in a unified framework. However, Zhao *et al.* [54] show that learning an invariant representation, in addition to possibly ignoring signals that can be important for new domains, is not enough to guarantee target generalization. Furthermore, as evidenced by the strong performance of ERM [15], these methods are either too strong to optimize reliably or too weak to achieve their goals [51].

Data Augmentation for Domain Generalization: Another effective strategy to address domain generalization [15, 47] is by using data augmentation. These methods focus on manipulating the inputs to assist in learning general representations. Zhou *et al.* [55] use domain information for creating an additive noise to increase the diversity of training data distribution while preserving the semantic information of data. Yan *et al.* [48] use mixup to blend examples from the different training distributions. In RSC [19], the authors iteratively discard the dominant features from the training data, aiming to improve generalization. This approach is inspired by the style transfer literature, where the feature statistics encode domain-related information. Similarly, MixStyle [56] synthesizes novel domains by mixing the feature statistics of two instances. SagNets [34] propose disentangling style encodings from class categories to prevent style-biased predictions and focus more on the contents. The performance of these methods depends on whether the augmentation can help the model to learn invariance in the data.

In this work, we build upon observations from [15, 47, 52], which show that data augmentation plays a vital role in improving out-of-distribution generalization. Our approach uses uniform sampling and a rejection reward, inspired by TrivialAugment [33] and TeachAugment [42], respectively. This combination leads to the proposal of an effective data augmentation strategy for domain generalization.

3. Revisiting Random Data Augmentation for Domain Generalization

3.1. Problem Definition

We study the problem of Multi-Source Domain Generalization for classification. During training, we assume access to N datasets containing examples about the same task but collected under a different domain or environment, $\mathcal{D} = \{1, 2, .., N\}$. Let $\mathcal{S}$ be a training dataset containing samples from all training domains, $\mathcal{S} = \{(x_1, y_1, d_1), (x_2, y_2, d_2) \ldots, (x_M, y_M, d_M)\}$, with $M = |\mathcal{S}|$. Here, $x_i \in \mathcal{X}$ refers to an image, $y_i \in \mathcal{Y}$ is the class label, and $d_i \in \mathcal{D}$ is the domain label. Then, the goal of the domain generalization task is to learn a mapping $f_\theta \colon \mathcal{X} \to \mathcal{Y}$ parametrized by θ that generalizes well to an unseen domain, $\hat{d} \notin \mathcal{D}$. In addition, we also consider a domain classifier $h_\phi \colon \mathcal{X} \to \mathcal{D}$ parametrized by ϕ that learns to recognize the domain of a given sample from $\mathcal{S}$. As a baseline optimization problem, we consider the simple empirical risk minimization (ERM), which minimizes the average loss over all samples, $\theta^* = \arg\min_\theta \frac{1}{M} \sum_{(x,y)\in\mathcal{S}} \mathcal{L}(f_\theta(x), y)$, where $\mathcal{L}(\cdot)$ is the cross-entropy loss function.

3.2. Data Augmentation Search Space

A well-known approach to achieving domain generalization is transforming the training samples during the learning process to gain robustness against unseen domains [8, 33]. These transformations come from a predefined set of possible data augmentations that operate within a given range of magnitudes. We consider as standard transformations $\mathcal{T}_{weak} \colon \mathcal{X} \to \mathcal{X}_{weak}$ the random flip, crop, and slight color-jitter augmentations that are safe, i.e., do not destroy image semantics. Such weak transformations are used in every training step. On top of standard transformation, we may also apply more transformations selected from either data augmentation search space Default from RandAugment [9] or Wide from TrivialAugment (TA) [33]. Here, we use geometric transformations (ShearX/Y, TranslateX/Y, Rotate) and color-based transformations (Posterize, Solarize, Contrast, Color, Brightness, Sharpness, AutoContrast, Equalize, and Grey). However, unlike TA, we expand the magnitude ranges and construct $\mathcal{T}_{wider} \colon \mathcal{X}_{weak} \to \mathcal{X}_{wider}$ to include more aggressive data augmentation (see 6 from supplementary materials). For sampling transformations, we follow the TA procedure, which involves randomly sampling an operation and magnitude from the search space for each image.

3.3. Motivation for Wider Range

Random augmentation over a set of predefined transformations, as in TA, despite being very simple, is competitive to the state-of-the-art Data Augmentation in standard in-domain settings. In Table 1, we consider the performance of TA, RandAugment, and AutoAugment for domain generalization. As can be seen, such DA methods are already improving over ERM [44]. However, for domain generalization, we expect that more aggressive transformations can push the representation outside the training domains and help to adapt to new domains. As shown in Table 1, the uniform sampling strategy of TA, but with wider transformations, further improves over the rest of the methods.

Method	Search Space	Dataset					
		PACS	VLCS	OfficeHome	TerraInc	DomainNet	**Avg.**
ERM as in [44]	Weak	$84.2_{\pm0.1}$	$77.3_{\pm0.1}$	$67.6_{\pm0.2}$	$47.8_{\pm0.6}$	$44.0_{\pm0.1}$	64.2
RandAugment [9]	Default	$86.1_{\pm0.8}$	$78.7_{\pm0.7}$	$67.8_{\pm0.4}$	$44.7_{\pm1.4}$	$44.0_{\pm0.2}$	64.3
TA [33]	Wide	$85.5_{\pm1.1}$	$78.6_{\pm0.5}$	$68.0_{\pm0.2}$	$47.8_{\pm1.6}$	$43.8_{\pm0.2}$	64.7
AutoAugment [8]	Default	$85.8_{\pm0.5}$	$78.7_{\pm0.8}$	$68.4_{\pm0.2}$	$48.0_{\pm1.3}$	$43.7_{\pm0.2}$	64.9
TA (Ours)	Wider	$85.6_{\pm0.8}$	$78.6_{\pm0.4}$	$68.9_{\pm0.4}$	$48.3_{\pm0.8}$	$43.7_{\pm0.3}$	65.0

Table 1. Different strategies of data augmentation. We compare different search space ranges traditionally used and wider ones. TA with a wider search space leads to better average out-of-domain accuracy. Our experiments are repeated three times. For details about datasets and training procedures (see section 5).
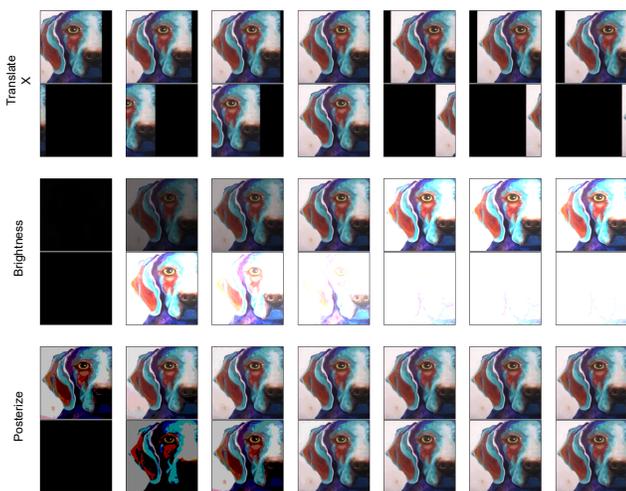

Figure 2. Sample transformations from TA with wide and wider search space on PACS dataset. For each transformation, the first row shows the range of transformed samples with wide search space and the second row with wider. We see that the wider space can lead to more variety but also extreme and detrimental transformations that do not keep the semantics of the image. This motivates us to use wider transformations but find a way to reject the extreme ones.

However, as shown in Figure 2, stronger augmentations can easily lead to extreme transformations that do not keep the semantics of the image. Thus, this work aims to improve this strong baseline by proposing a mechanism to reject those extreme augmentations. For more details about the used datasets and the training procedures, see section 5.

4. Rejecting Extreme Transformations

For each given input, we generate a weakly augmented version using standard transformation (i.e., using only a flip and a crop and slight color-jitter) and an enormously augmented version using $\mathcal{T}_{wider}$ transformation as defined

in the previous section. We then define a reward function $R(x, z)$ that, given an input x and metadata, either domain label d or class label y, provides a measurement of the quality for the transformed sample. Then, maximizing such a reward function allows selecting which augmentation is more suitable for the training:

$$\tilde{x} = \begin{cases} \mathcal{T}_{wider}(x) & \text{if } R(\mathcal{T}_{wider}(x), z) \geq R(\mathcal{T}_{weak}(x), z) \\ \mathcal{T}_{weak}(x) & \text{otherwise} \end{cases}$$
(1)

4.1. Augmentation Reward

Intuitively, for domain generalization, a good augmentation creates challenging samples that provide useful training information without losing the sample's original meaning (i.e., the sample's class). We use the teacher-student paradigm to achieve this goal and introduce a unified reward function consisting of diversity and semantic consistency components for selecting the appropriate augmentation. Considering $\tilde{x}$ as an augmented sample, the reward function is defined as:

$$R(\tilde{x}, z) = (1 - \lambda)R_{div}(\tilde{x}, z) - \lambda R_{con}(\tilde{x}, z)$$
(2)

where λ is the balancing coefficient between diversity and consistency. Here, z refers to either the domain of the sample d or the class label y, and it is specified in the following sections for every term of the proposed reward. In the previous equation, the R_{div} term enforces diversity in the data by exploring the augmentations of the input, while R_{con} keeps the semantic meaning of augmented sample $\tilde{x}$.

4.2. Diverse Student and Consistent Teacher

To make our idea work, we must ensure that the diversity reward considers the latest changes in the model. Thus, as a reward for diversity, we use the cross-entropy loss $\mathcal{L}$ of a classifier h with parameters ϕ trained to detect the domain of the image x:
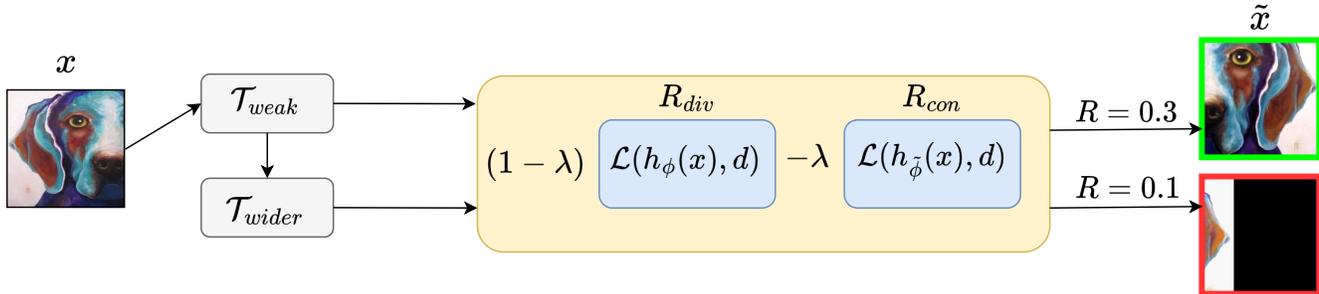
$$R_{div}(x, d) = \mathcal{L}(h_\phi(x), d)$$
(3)

Figure 3. Overview of DCAug^{domain} procedure for rejecting extreme augmentations. After calculating R_{div} and R_{con} for $\mathcal{T}_{weak}$ and $\mathcal{T}_{wider}$ our method selects the transformation with the highest reward (green box). It updates the label classifier f_θ and domain student h_ϕ using the transformed input $\tilde{x}$. DCAug^{label} and TeachDCAug^{label} also follow the same procedure by replacing d and h_ϕ by y and f_θ respectively (see 10 from supplementary materials for more visual changes of the selected images).

Algorithm 1 DCAug Training Procedure
_____
Input: source domains S, label classifier f_θ, domain clas-
     sifier h_ϕ, transformations $\mathcal{T}_{weak}$ and $\mathcal{T}_{wider}$, learning
     rate η.
Output: label classifier f_θ or $f_{\tilde{\theta}}$.
  1: **for** minibatch (x, y, d) in training dataset S **do**
  2:     $\hat{x}_1 \leftarrow \mathcal{T}_{weak}(x)$
  3:     $\hat{x}_2 \leftarrow \mathcal{T}_{wider}(\hat{x}_1)$
  4:     select $\hat{x}$ according to Eq. 1
  5:     $\theta \leftarrow \theta - \eta\nabla_\theta[\mathcal{L}(f_\theta(\hat{x}, y)]$
  6:     **if** DCAug^{label} **then**
  7:        $\tilde{\theta} = (1 - \beta)\theta + \beta\tilde{\theta}$
  8:     **else if** DCAug^{domain} **then**
  9:        $\phi \leftarrow \phi - \eta\nabla_\phi[\mathcal{L}(h_\phi(\hat{x}, d)]$
10:       $\tilde{\phi} = (1 - \beta)\phi + \beta\tilde{\phi}$
11:     **end if**
12: **end for**
_____

In this way, the reward would avoid favoring multiple times the same samples because they are already included in the model. At the same time, the consistency reward needs to be robust because we must classify those samples correctly. To do that, an exponential moving average (EMA) of our domain classifier $\tilde{\phi}$ is used as a consistent teacher:

$$\tilde{\phi} = (1 - \beta)\phi + \beta\tilde{\phi}$$
$$R_{con}(x, d) = \mathcal{L}(h_{\tilde{\phi}}(x), d) \quad (4)$$

where β defines the smoothness of the moving average and is fixed at 0.999 for all experiments, we call this approach DCAug^{domain}.

Alternatively, in situations where the domain meta-data d is unavailable, we can rewrite Eqs. 3 and 4 by using the label classifier f_θ as the teacher and student, and the class label as ground truth. The method is referred to hereafter as

DCAug^{label} and uses the rewards terms:

$$R_{div}(x, y) = \mathcal{L}(f_\theta(x), y)$$
$$R_{con}(x, y) = \mathcal{L}(f_{\tilde{\theta}}(x), y) \quad (5)$$

being $\tilde{\theta}$ the exponential moving average (EMA) of θ. This method also allows us to use $\tilde{\theta}$ instead of θ as the final classifier, which usually results in a more robust model [3]. We call this special variant TeachDCAug^{label}. Figure 3 shows an overview of the DCAug procedure. For each iteration, our training schema comprises two phases. In the first phase, we freeze θ, ϕ parameters and select the most appropriate transformation based on our reward function R. In the second phase, we update θ, ϕ using a gradient descent procedure. The full algorithm of the DCAug training procedure is presented in Algorithm 1.

5. Experiments

Dataset. Following DomainBed benchmark [15], we evaluate our method on five diverse datasets. PACS [25] is a 7-way object classification task with 4 domains and 9,991 samples. VLCS [11] is a 5-way classification task with 4 domains and 10,729 samples. This dataset mostly contains real photos. The distribution shifts are subtle and simulate real-life scenarios well. OfficeHome [45] is a 65-way classification task depicting everyday objects with 4 domains and a total of 15,588 samples. TerraIncognita [4] is a 10-way classification problem of animals in wildlife cameras, where the 4 domains are different locations. There are 24,788 samples. This represents a realistic use case where generalization is indeed critical. DomainNet [36] is a 345-way object classification task with 6 domains. With a total of 586,575 samples, DomainNet is larger than most of the other evaluated datasets in both samples and classes.

Evaluation protocols and implementation details. All performance scores are evaluated by leave-one-out cross-validation, averaging all cases that use a single domain as

Method	Category	Dataset					
		PACS	VLCS	OfficeHome	TerraInc	DomainNet	**Avg.**
ERM [44]	*Baseline*	$84.2_{\pm0.1}$	$77.3_{\pm0.1}$	$67.6_{\pm0.2}$	$47.8_{\pm0.6}$	$44.0_{\pm0.1}$	64.2
MMD [26]		$84.7_{\pm0.5}$	$77.5_{\pm0.9}$	$66.4_{\pm0.1}$	$42.2_{\pm1.6}$	$23.4_{\pm9.5}$	58.8
IRM [2]		$83.5_{\pm0.8}$	$78.6_{\pm0.5}$	$64.3_{\pm2.2}$	$47.6_{\pm0.8}$	$33.9_{\pm2.8}$	61.6
GroupDRO [39]	*Domain-Invariant*	$84.4_{\pm0.8}$	$76.7_{\pm0.6}$	$66.0_{\pm0.7}$	$43.2_{\pm1.1}$	$33.3_{\pm0.2}$	60.7
DANN [12]		$83.7_{\pm0.4}$	$78.6_{\pm0.4}$	$65.9_{\pm0.6}$	$46.7_{\pm0.5}$	$38.3_{\pm0.1}$	62.6
CORAL [41]		$86.2_{\pm0.3}$	$78.8_{\pm0.6}$	$68.7_{\pm0.3}$	$47.6_{\pm1.0}$	$41.5_{\pm0.1}$	64.5
mDSDI [6]		$86.2_{\pm0.2}$	$79.0_{\pm0.3}$	$69.2_{\pm0.4}$	$48.1_{\pm1.4}$	$42.8_{\pm0.2}$	65.1
DDAIG [55]		83.1	-	65.5	-	-	-
MixStyle [56]		$85.2_{\pm0.3}$	$77.9_{\pm0.5}$	$60.4_{\pm0.3}$	$44.0_{\pm0.7}$	$34.0_{\pm0.1}$	60.3
RSC [19]		$85.2_{\pm0.9}$	$77.1_{\pm0.5}$	$65.5_{\pm0.9}$	$46.6_{\pm1.0}$	$38.9_{\pm0.5}$	62.7
Mixup [48]	*Data Augmentation*	$84.6_{\pm0.6}$	$77.4_{\pm0.6}$	$68.1_{\pm0.3}$	$47.9_{\pm0.8}$	$39.2_{\pm0.1}$	63.4
SagNets [34]		$86.3_{\pm0.2}$	$77.8_{\pm0.5}$	$68.1_{\pm0.1}$	$48.6_{\pm1.0}$	$40.3_{\pm0.1}$	64.2
DCAug^{domain} (Ours)		$86.1_{\pm0.9}$	$78.9_{\pm0.5}$	$68.8_{\pm0.4}$	$48.7_{\pm0.8}$	$43.7_{\pm0.3}$	65.2
DCAug^{label} (Ours)		$86.1_{\pm0.7}$	$78.6_{\pm0.4}$	$68.3_{\pm0.4}$	$49.3_{\pm1.5}$	$43.8_{\pm0.2}$	65.2
TeachDCAug^{label} (Ours)		$88.4_{\pm0.2}$	$78.8_{\pm0.4}$	$70.4_{\pm0.2}$	$51.1_{\pm1.1}$	$46.4_{\pm0.1}$	**67.0**

Table 2. Comparison with domain generalization methods Out-of-domain accuracies on five domain generalization benchmarks are presented. We highlight the best overall result. For each category, we also report the average accuracy per dataset. Accuracies other than our methods (DCAug) are from [7, 15]. Our experiments are repeated three times.

the target (test) domain and the others as the source (training) domains. We employ DomainBed training and evaluation protocols [15]. In particular, for training, we use ResNet-50 [17] pre-trained on the ImageNet [38] as the default backbone. The model is optimized using Adam [21] optimizer. A mini-batch contains all domains and 32 examples per domain. For the model hyperparameters, such as learning rate, dropout rate, and weight decay, we use the same configuration as proposed in [7]. We follow [7] and train models for 15000 steps on DomainNet and 5000 steps for other datasets, corresponding to a variable number of epochs dependent on dataset size. Every experiment is repeated three times with different seeds. We leave 20% of source domain data for validation. We use training-domain validation for the model selection, in which, for each random seed, we choose the model, maximizing the accuracy of the validation set. The balancing coefficient of our method, λ, is coarsely tuned on the validation with three different values: [0.2, 0.5, 0.8].

5.1. Main Results

In this section, we compare three variations of our model, DCAug^{domain}, DCAug^{label} and TeachDCAug^{label} with and without domain meta-data (as explained in section 4.2), with 11 related methods in DG. Those methods are divided into two families: data augmentation and domain-invariant representation. For data augmentation, we compare with Mixup [48], MixStyle [56], DDAIG [55], SagNets

[34] and RSC [19]. For invariant representation learning, we compare with IRM [2], GroupDRO [39], CORAL [41], MMD [26], DANN [12] and mDSDI [6]. We also included ERM as a strong baseline as shown in [15].

Table 2 shows the overall performance of DCAug and other methods on five domain generalization benchmarks on a classification task. The full result per dataset and the domain are provided in the supplementary material. From the table, we observe that, as shown in [20], most methods struggle to reach the performance of a simple ERM adapted to multiple domains, and only a few methods manage to obtain good results on all datasets. TeachDCAug^{label}, which essentially is the moving average version of DCAug^{label}, while being simple, manages to rank among the first on all datasets, outperforming all data augmentation and domain-invariant methods. Furthermore, DCAug^{domain} and DCAug^{label} manage to outperform all data augmentation methods and obtain comparable results with the best domain-invariant methods which underlines the importance of good data augmentation for DG. Another important observation is that as the dataset size increased, from PACS with 9,991 samples to DomainNet with 586,575 samples (see section 5), most of the methods, especially from the domain-invariant family struggled to show any performance increase compared to the ERM baseline. This poor performance is probably due to the large size and larger number of domains of TerraIncognita and DomainNet which makes all approaches that try to artificially increase the general-

ization of the algorithm not profitable. On the other hand, our approach manages to keep the performance close or better (in the case of TerraIncognita) to ERM. Recently, a new method based on ensembling models [37] trained with different hyperparameters has reached an average accuracy of 68 on the evaluated datasets. However, this approach does not belong to the studied methods and is orthogonal to them. Compared to ERM, DCAug has a small additional computational cost. In particular, DCAug^{domain}, other than updating the parameters of both domain and label classifier for each sample, computes the loss of the domain classifier twice without the need to calculate the gradients (see C from supplementary materials for full characterization.)
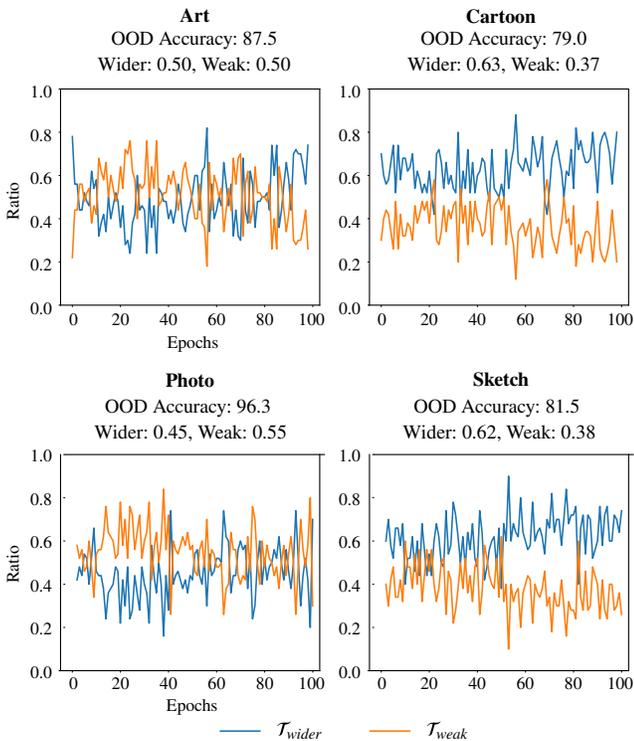
5.2. Empirical analysis of DCAug

Figure 4. Evolution over epochs of Weak vs Wider augmentations on the four domains of the PACS dataset. The title of each plot shows the domain, out-of-domain accuracies, and the average ratio of each augmentation for the entire training run. In the plots, we see that for Cartoon and Sketch, the two domains that are farther from the pre-trained model on ImageNet and with lower performance, strong transformations are preferred over weak ones.

Rejection rate of strong augmentations. We study the evolution of the rejection rate of strong augmentations over epochs on the PACS dataset. We use the same balancing hyperparameter λ as in the main experiments. For each domain, we show the selection of weak and strong augmentations for the entire training. As we observe in Figure 4, the

rejecting rate is domain-dependent. As models have been pre-trained on ImageNet [38], we can see that our method selects weak and strong augmentations equally for domains that are closer to the pre-trained dataset (Art and Photo). However, for Cartoon and Sketch, which are far from ImageNet, we observe that our method uses more strong augmentations. This observation is in line with other works that suggest the effect of data augmentations diminishes if the training data already covers most of the variations in the dataset [31, 47].

Measuring diversity and consistency. Following [14], we compare the diversity and consistency (affinity) of the transformation space that our proposed approach induces with TA (Wide), TA (Wider), and ERM. Intuitively, consistency measures the level of distortion caused by a given data augmentation schema on the target dataset. Here, in our case, we use the in-domain validation and out-domain test set to measure in and out-domain performance, respectively. On the other hand, diversity is a model-dependent element and captures the difficulty of the model to fit the augmented training data (see D from supplementary materials for precise definitions). As shown in Figure 5, neither of the two extremes, TA (Wider) as the most diverse and ERM with standard data augmentations as the most consistent, is sufficient for the best final performance. However, all of our proposed approaches provide a good trade-off between consistency and diversity, which results in the best absolute performance for both in-domain and out-domain.

In-domain performance. The in-domain performance of Autoaugment, RandAugment, TA (Wide), TA (Wider) is investigated, and our proposed methods are in Table 3. It has the exact opposite ranking compared to Table 1, which shows that aggressive data augmentation can indeed harm the in-domain performance. Our method, however, has proven to be highly effective in both cases, demonstrating that by limiting the scope of transformation, we can achieve optimal outcomes that combine the best of both worlds.

Diverse domain and consistent label. Given that our final target is to select a transformation that creates challenging samples with diverse domains without losing the sample's original meaning, one might want to use the domain classifier and label classifier to satisfy this goal. In particular, we can write a variation of our reward functions as follows:

$$\begin{aligned} R_{div}(x, d) &= \mathcal{L}(h_\phi(x), d) \\ R_{con}(x, y) &= \mathcal{L}(f_\theta(x), y), \end{aligned} \quad (6)$$

in which the diversity is measured in terms of domain labels d, and the consistency in class labels y. This formulation resembles the reward function used in [55], although for different purposes. We also derive another variation to this reward function, which uses the EMA version of each model. Also, for these rewards, we tune the hyperparameter λ to
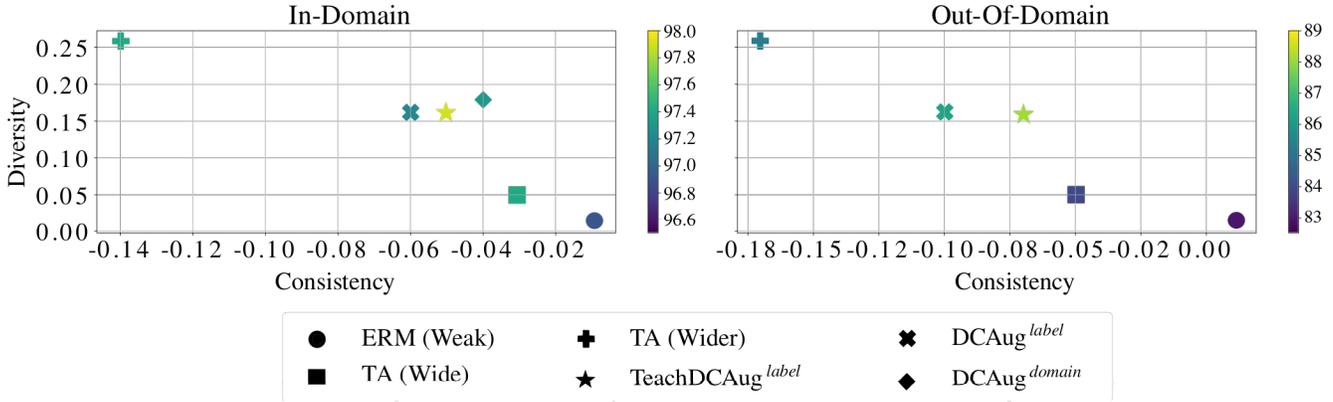
Figure 5. Consistency and diversity for different methods for in-domain (left) and out-of-domain (right) settings on the PACS dataset. Color represents the classification accuracy on the test set. For high accuracy, we need a good trade-off between diversity and consistency.

Method	Search Space	Dataset					
		PACS	VLCS	OfficeHome	TerraInc	DomainNet	Avg.
AutoAugment [8]	Default	$97.3_{\pm0.2}$	$87.0_{\pm0.1}$	$82.9_{\pm0.3}$	$90.1_{\pm0.2}$	$62.2_{\pm0.1}$	83.9
TA (Ours)	Wider	$97.6_{\pm0.1}$	$87.2_{\pm0.1}$	$83.4_{\pm0.3}$	$89.7_{\pm0.1}$	$62.0_{\pm0.1}$	84.0
RandAugment [9]	Default	$97.3_{\pm0.2}$	$87.1_{\pm0.2}$	$82.8_{\pm0.3}$	$91.2_{\pm0.1}$	$62.4_{\pm0.4}$	84.2
TA [33]	Wide	$97.6_{\pm0.2}$	$87.2_{\pm0.1}$	$83.8_{\pm0.4}$	$90.9_{\pm0.2}$	$62.7_{\pm0.1}$	84.4
DCAug^{domain} (Ours)	Wider	$97.5_{\pm0.2}$	$87.1_{\pm0.1}$	$83.5_{\pm0.3}$	$91.6_{\pm0.2}$	$62.8_{\pm0.1}$	84.5
DCAug^{label} (Ours)	Wider	$97.6_{\pm0.2}$	$87.3_{\pm0.2}$	$83.4_{\pm0.4}$	$91.3_{\pm0.2}$	$62.8_{\pm0.1}$	84.5
TeachDCAug^{label} (Ours)	Wider	$98.1_{\pm0.2}$	$87.5_{\pm0.2}$	$84.1_{\pm0.4}$	$92.5_{\pm0.1}$	$65.6_{\pm0.1}$	**85.6**

Table 3. In-domain accuracies of our methods on five domain generalization benchmarks. Our experiments are repeated three times.

find the best balance between diversity and consistency. As reported in Table 4, these two formulation variants are ineffective. In particular, since the task of domain classification is significantly easier than the target task, finding a good balance between these two turns out to be difficult [40].

6. Conclusion

In this work, a method is introduced to improve the performance of data augmentation when multiple domains are

Diversity	Consistency	OOD Accuracy
$\mathcal{L}(h_\phi(x), d)$	$\mathcal{L}(f_\theta(x), y)$	81.8
$\mathcal{L}(h_{\tilde{\phi}}(x), d)$	$\mathcal{L}(f_{\tilde{\theta}}(x), y)$	80.1
$\mathcal{L}(f_\theta(x), y)$	$\mathcal{L}(f_{\tilde{\theta}}(x), y)$	**86.1**
$\mathcal{L}(h_\phi(x), d)$	$\mathcal{L}(h_{\tilde{\phi}}(x), d)$	**86.1**

Table 4. Variations of our reward function on PACS dataset. $h_{\tilde{\phi}}$ and $f_{\tilde{\theta}}$ refer to the EMA version of their corresponding models. Our methods are highlighted in the table.

available at training time, yet the distribution of the test domain is different from those and unknown. In this setting, we show that the state-of-the-art in-domain augmentation of TrivialAugment [33] based on uniform sampling of predefined transformation is beneficial, helping to improve results for a baseline based on ERM, which is effective for domain generalization [15]. Then, we propose to further improve results by increasing the magnitude of those transformations while preserving the random sampling. This makes sense for domain generalization as we want to learn an unknown domain. Finally, we propose a rejection scheme that removes extreme and harmful transformations during training based on a reward function that compares the performance of a label classifier with its exponential moving average. These contributions allowed our method to achieve equal or better performance than state-of-the-art methods on five challenging domain generalization datasets with a minimum intervention in the standard ERM pipeline.

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. 2

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3, 6, 16, 17, 18

[3] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022. 5

[4] 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. 5, 12

[5] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011. 1

[6] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201, 2021. 3, 6, 16, 17, 18

[7] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021. 6, 12, 15

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 1, 2, 3, 4, 8

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 1, 2, 3, 4, 8, 13

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 15

[11] 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. 5, 12

[12] 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. 2, 6, 16, 17, 18

[13] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1055–1064, 2021. 2

[14] Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*, 2020. 7, 15

[15] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 3, 5, 6, 8, 12, 15

[16] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 1–16. Springer, 2020. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[18] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019. 2

[19] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020. 3, 6, 16, 17, 18

[20] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022. 6

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2021. 1

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[25] 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. 5, 12

[26] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5400–5409, 2018. 2, 6, 16, 17, 18

[27] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2

[28] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada: Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020. 2

[29] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[30] Tom Ching LingChen, Ava Khonsari, Amirreza Lashkari, Mina Rafi Nazari, Jaspreet Singh Sambee, and Mario A Nascimento. Uniformaugment: A search-free probabilistic data augmentation approach. *arXiv preprint arXiv:2003.14348*, 2020. 2

[31] Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Rong Jin, Xiangyang Ji, and Antoni B Chan. An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation. *arXiv preprint arXiv:2205.12753*, 2022. 7

[32] Saypraseuth Mounsaveng, Issam Laradji, Ismail Ben Ayed, David Vazquez, and Marco Pedersoli. Learning data augmentation with online bilevel optimization for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1691–1700, 2021. 2

[33] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 774–782, 2021. 1, 2, 3, 4, 8, 13

[34] 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. 3, 6, 16, 17, 18

[35] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 1

[36] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 5, 12

[37] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *arXiv preprint arXiv:2205.09739*, 2022. 7

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 6, 7

[39] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 3, 6, 16, 17, 18

[40] Amrith Setlur, Don Dennis, Benjamin Eysenbach, Aditi Raghunathan, Chelsea Finn, Virginia Smith, and Sergey Levine. Bitrate-constrained dro: Beyond worst case robustness to unknown group shifts. *arXiv preprint arXiv:2302.02931*, 2023. 8

[41] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016. 3, 6, 16, 17, 18

[42] Teppei Suzuki. Teachaugment: Data augmentation optimization using teacher knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10904–10914, 2022. 1, 2, 3

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[44] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 1, 3, 4, 6, 16, 17, 18

[45] 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. 5, 12

[46] Longhui Wei, An Xiao, Lingxi Xie, Xiaopeng Zhang, Xin Chen, and Qi Tian. Circumventing outliers of autoaugment with knowledge distillation. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 2

[47] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021. 3, 7

[48] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020. 3, 6, 16, 17, 18

[49] Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022. 2

[50] Zihan Yang, Richard O Sinnott, James Bailey, and Qiuhong Ke. A survey of automated data augmentation algorithms for deep learning-based image classication tasks. *arXiv preprint arXiv:2206.06544*, 2022. 1, 2

[51] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pages 26397–26411. PMLR, 2022. 3

[52] Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie Harmon, Baris Turkbey, Holger Roth, Andriy Myronenko, Daguang Xu, and Ziyue Xu. When unseen domain generalization is unnecessary? rethinking data augmentation. *arXiv preprint arXiv:1906.03347*, 2019. 3

[53] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*, 2019. 2

[54] 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. 3

[55] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032, 2020. 2, 3, 6, 7, 16, 17

[56] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 3, 6, 16, 17, 18