Modality-Agnostic Debiasing for Single Domain Generalization

Sanqing Qu[1*], Yingwei Pan[2], Guang Chen[1], Ting Yao[2], Changjun Jiang[1], Tao Mei[2]
[1]Tongji University, [2]JD.com
{2011444, guangchen, cjjiang}@tongji.edu.cn, {panyw.ustc, tingyao.ustc}@gmail.com, tmei@live.com

Abstract

Deep neural networks (DNNs) usually fail to generalize well to outside of distribution (OOD) data, especially in the extreme case of single domain generalization (single-DG) that transfers DNNs from single domain to multiple unseen domains. Existing single-DG techniques commonly devise various data-augmentation algorithms, and remould the multi-source domain generalization methodology to learn domain-generalized (semantic) features. Nevertheless, these methods are typically modality-specific, thereby being only applicable to one single modality (e.g., image). In contrast, we target a versatile Modality-Agnostic Debiasing (MAD) framework for single-DG, that enables generalization for different modalities. Technically, MAD introduces a novel two-branch classifier: a biased-branch encourages the classifier to identify the domain-specific (superficial) features, and a general-branch captures domain-generalized features based on the knowledge from biased-branch. Our MAD is appealing in view that it is pluggable to most single-DG models. We validate the superiority of our MAD in a variety of single-DG scenarios with different modalities, including recognition on 1D texts, 2D images, 3D point clouds, and semantic segmentation on 2D images. More remarkably, for recognition on 3D point clouds and semantic segmentation on 2D images, MAD improves DSU by 2.82% and 1.5% in accuracy and mIOU.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable success in various tasks under the assumption that training and testing domains are independent and sampled from identical or sufficiently similar distribution [2, 48]. However, this assumption often does not hold in most real-world scenarios. When deploying DNNs to unseen or out-of-distribution (OOD) testing domains, inevitable performance degeneration is commonly observed. The difficulty mainly originates from that the backbone of DNNs extracts more domain-specific (superficial) features together with domain-generalized (semantic) features. Therefore, the classifier is prone to paying much attention to those domain-specific features, and learning unintended decision rule [53]. To mitigate this issue, several appealing solutions have been developed, including *Domain Adaptation (DA)* [18, 32, 36, 40, 41] and *Domain Generalization (DG)* [31, 56, 62, 65]. Despite showing encouraging performances on OOD data, their real-world applications are still limited due to the requirement to have the data from other domain (i.e., the unseen target domain or multiple source domains with different distributions). In this work, we focus on an extreme case in domain generalization: *single domain generalization (single-DG)*, in which DNNs are trained with single source domain data and then required to generalize well to multiple unseen target domains.

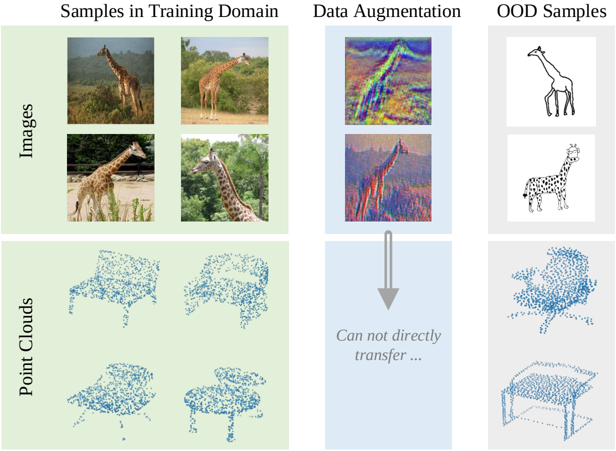


Figure 1. Most existing single-DG techniques devise various data augmentation algorithms to introduce various image textures and styles, pursuing the learning of domain-generalized features. However, these approaches are modality-specific, and only applicable to single modality (e.g., image). Hence it is difficult to directly employ such single-DG approach for 3D point clouds, since the domain shifts in 3D point clouds only reflect the geometric differences rather than texture and style differences.

Previous researches [19,55] demonstrate that the specific local textures and image styles tailored to each domain are two main causes, resulting in domain-specific features for images. To alleviate this, recent works [30, 37, 58, 63] de-

[*]This work was performed at JD.com

sign a variety of data-augmentation algorithms to introduce diversified textures and image styles. The DG methodologies are then remolded with these data-augmentation algorithms to facilitate the learning of domain-generalized features. Nevertheless, such solution for single-DG is typically modality-specific and only applicable to the single modality inputs of images. When coming a new modality (e.g. 3D point clouds), it is difficult to directly apply these techniques to tackle single-DG problem. This is due to the fact that the domain shift in 3D point clouds is interpreted as the differences of 3D structural information among multiple domains, instead of the texture and style differences in 2D images [10, 39]. Figure 1 conceptually illustrates the issue, which has been seldom explored in the literature.

In this paper, we propose to address this limitation from the standpoint of directly strengthening the capacity of classifier to identify domain-specific features, and meanwhile emphasize the learning of domain-generalized features. Such way completely eliminates the need of modality-specific data augmentations, thereby leading to a versatile modality-agnostic paradigm for single-DG. Technically, to materialize this idea, we design a novel Modality-Agnostic Debiasing (MAD) framework, that facilitates single domain generalization under a wide variety of modalities. In particular, MAD integrates the basic backbone for feature extraction with a new two-branch classifier structure. One branch is the biased-branch that identifies those superficial and domain-specific features with a multi-head cooperated classifier. The other branch is the general-branch that learns to capture the domain-generalized representations on the basis of the knowledge derived from the biased-branch. It is also appealing in view that our MAD can be seamlessly incorporated into most existing single-DG models with data-augmentation, thereby further boosting single domain generalization.

We analyze and evaluate our MAD under a variety of single-DG scenarios with different modalities, ranging from recognition on 2D images, 3D point clouds, 1D texts, to semantic segmentation on 2D images. Extensive experiments demonstrate the superior advantages of MAD when being plugged into a series of existing single-DG techniques with data-augmentation (e.g., Mixstyle [65] and DSU [30]). More remarkably, for recognition on point cloud benchmark, MAD significantly improves DSU in the accuracy from 33.63% to 36.45%. For semantic segmentation on image benchmark, MAD advances DSU with mIoU improvement from 42.3% to 43.8%.

2. Related Work

2.1. Domain Adaptation

Over the last decade, many efforts have been devoted to domain adaptation (DA) to address the OOD issue [18, 32, 57]. DA methods are developed to utilize the labeled source domain and the unlabeled out-of-distributed target domain in a transductive learning manner. Existing DA approaches can be briefly grouped into two paradigms, i.e., moment matching [14, 32, 47] and adversarial alignment [18, 25, 26]. DA methods have achieved significant progress in many applications, e.g., object recognition [35, 40, 41, 60], semantic segmentation [25, 47], and object detection [4, 11, 26]. Nevertheless, the requirement of both source and target domain data during training significantly limits their practical deployment. Besides, in DA manner, DNNs are typically coupled with source and target domains, affecting their capacity to generalize to other domains. In this work, we focus on a more challenging scenario where DNNs are required to generalize well to multiple unseen domains.

2.2. Domain Generalization

Different from DA, domain generalization (DG) expects to learn generalized DNNs with the assistance of multiple source domains [31, 56, 65], without the access of target domain. Currently, DG methods can mainly be categorized into three dimensions, including domain alignment, data augmentation/generation, and ensemble learning. Most existing DG methods [20, 27, 33] belong to the category of domain alignment. Their motivation is straightforward: features that are invariant to the source domain shifts should also be generalized to any unseen target domain shift. Data generation is another popular technique for DG [45, 50, 59]. The goal is to generate diverse and rich data to boost the generalization ability of DNNs. Existing methods typically remould the Variational Autoencoder (VAE) [28], and the Generative Adversarial Networks (GAN) [21] to execute diversified data generation. Ensemble learning [17] commonly learns multiple copies of the same model with different initialization and then utilizes their ensemble for final prediction. As the variant of ensemble learning, weight averaging [5], domain-specific neural networks [64], and domain-specific batch normalization [43] have recently achieved promising results. Nevertheless, it is non-trivial to directly apply these DG techniques for single domain generalization.

2.3. Single Domain Generalization

Single domain generalization (single-DG) is an extreme case of domain generalization, where DNNs are trained with only one source domain data and required to perform well to multiple unseen target domains. It is more challenging than DA and DG, yet indeed more realistic in practical applications. To address this challenging problem, several methods [30, 37, 58] have designed various data augmentation algorithms to enhance the diversity and informativeness of training data. In [58], the authors propose a style-complement module to synthesize images from di-

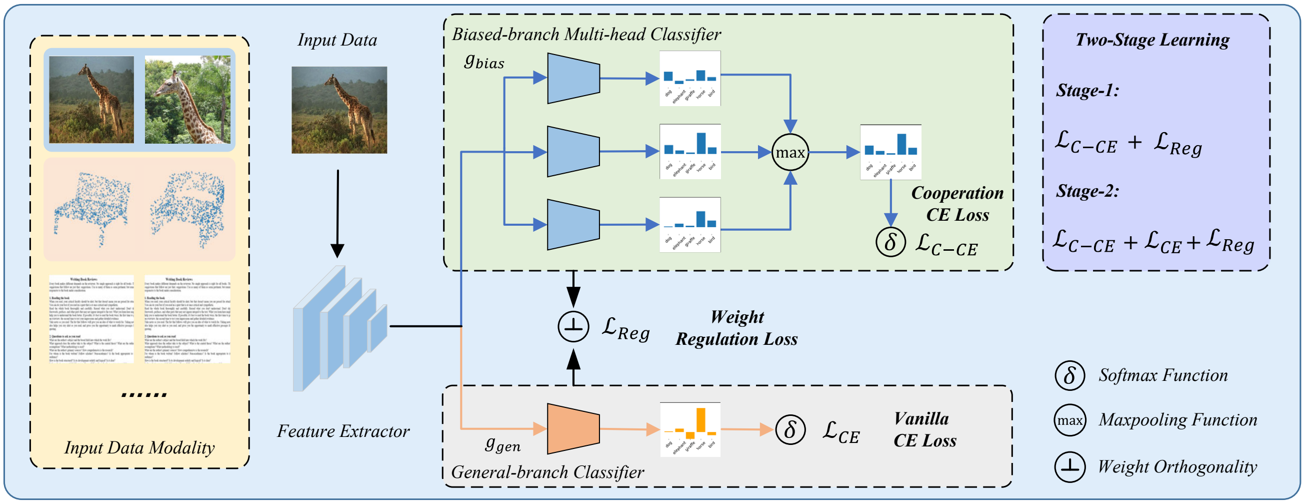

Input Data Modality
......
Input Data
Feature Extractor
Biased-branch Multi-head Classifier
g_{bias}
max
Cooperation CE Loss
δ $\mathcal{L}_{C-CE}$
$\perp$ $\mathcal{L}_{Reg}$
Weight Regulation Loss
g_{gen}
General-branch Classifier
δ $\mathcal{L}_{CE}$
Vanilla CE Loss
Two-Stage Learning
Stage-1:
$\mathcal{L}_{C-CE} + \mathcal{L}_{Reg}$
Stage-2:
$\mathcal{L}_{C-CE} + \mathcal{L}_{CE} + \mathcal{L}_{Reg}$
δ Softmax Function
max Maxpooling Function
$\perp$ Weight Orthogonality

Figure 2. An overview of our Modality-Agnostic Debiasing (MAD) framework. The main challenge for DNNs to realize single domain generalization (single-DG) is that the classifier tends to over-emphasize those domain-specific features, yielding unintended decision rules [53]. To address this challenge, we propose to strengthen the capability of classifier of identifying domain-specific features, and meanwhile emphasising the learning of domain-generalized features. Technically, the MAD framework integrates the basic backbone for feature extraction with a new two-branch classifier, i.e., the biased branch and the general branch. We implement MAD in a two-stage learning mechanism. In the first stage, the biased-branch is utilized to identify those domain-specific features with a multi-head cooperated classifier. In the second stage, the general-branch classifier is encouraged to capture those domain-generalized features on the basis of the knowledge from the biased-branch, i.e., with the guidance from $\mathcal{L}_{reg}$. Our framework is modality-agnostic and can be applied to various modalities such as images, point clouds, and texts.

verse distribution. In [30], synthesized feature statistics are introduced to model the uncertainty of domain shifts during training. To regulate single-DG training, [15] applies a variety of visual corruptions as augmentation and designs a new attention consistency loss. A novel image meta-convolution network is developed in [51] for capturing more domain-generalized features. Nevertheless, most methods are modality-specific and only applicable for images inputs. When we encounter a new data modality, they are commonly not available to deploy. The reason behind is that for different data modalities, domain shifts tend to be different. For example, the differences in 3D geometry structure among multiple domains are the origin of domain shifts for point clouds, instead of style and texture differences in 2D images. Our work delves into this limitation and targets for proposing a general and versatile framework for single-DG that is agnostic to data modality.

3. Methodology

3.1. Preliminary

We consider an extreme case in generalization: single domain generalization (single-DG), where the goal is to train DNNs with single source domain $\mathcal{D}_S$ that perform well to multiple unseen target domains: $\{\mathcal{D}_T^1, \mathcal{D}_T^2, \ldots, \mathcal{D}_T^Z\}$. In particular, we consider the K-way classification. We denote $\mathcal{D}_S = \{(x_i, y_i)\}_{i=1}^n$, where $x \in \mathcal{X} \subset \mathbb{R}^X, y \in \mathcal{Y} \subset \mathbb{R}^K$. The whole DNN architecture is represented as $F = g \circ f$, where $f : \mathbb{R}^X \to \mathbb{R}^D$ denotes the feature extractor and $g : \mathbb{R}^D \to \mathbb{R}^K$ is the classifier. This setting is guaranteed under a general assumption in domain generalization: There are domain-generalized features e_g in the domain $\mathcal{D}_S$ whose correlation with label is consistent across domains, and domain-specific features e_s whose correlation with label varies across domains. Classifiers that rely on domain-generalized features e_g perform much better on new unseen domains than those that depend on domain-specific features e_s. In this setting, directly applying the vanilla empirical risk minimization (ERM) [48] on $\mathcal{D}_S$ commonly results in a sub-optimal model that does not generalize well to unseen domains. The main reason originates from that the feature extractor f often extracts more domain-specific features e_s together with domain-generalized features e_g [53]. DNNs trained with SGD often count on the simplest features [44], which leads to a tendency for the classifier g to overemphasize e_s and pay less attention to e_g, resulting in unintended decision rules.

Prior methods [15,30,37,58] have designed various data-augmentation algorithms to encourage the feature extractor g to learn more domain-generalized features e_g and suppress those domain-specific features e_s. However, these algorithms are typically modality-specific, and largely limited to images. Instead, we propose to mitigate this limitation by directly strengthening the capacity of classifier for identifying domain-specific features, and meanwhile emphasising the learning of domain-generalized features. That completely eliminates the requirement of modality-specific data augmentations, pursuing a versatile and modality-agnostic paradigm for single-DG. Technically, we present a novel modality-agnostic debiasing (MAD) framework. MAD integrates the basic backbone for feature extraction with a new two-branch classifier structure. One branch is the biased-branch that identifies those domain-specific features e_s with a multi-head cooperated classifier. The other is that learns to capture the domain-generalized features e_g with the knowledge derived from the biased-branch. Figure 2 illustrates the detailed architecture of our MAD.

3.2. Identifying Domain-specific Features

There have been some efforts [6, 12, 38, 52] in domain generalization to realize domain-specific features e_s and domain-generalized features e_g separation. Nevertheless, most of them require multiple training domains and predefined domain labels, making them inapplicable for single-DG. Moreover, [38] has pointed out that given a trained classifier, it is non-trivial to uniquely decompose the classifier weight into domain-specific and domain-generalized terms, especially with only one source domain data.

To alleviate these issues, we propose a simple yet effective domain-specific feature identification strategy. Our motivation is straightforward: since the vanilla classifier trained with SGD will inadvertently focus more on those domain-specific features, the weights of the trained classifier can be considered as an indicator of those features.

Nevertheless, a single vanilla classifier is typically not effective to locate all domain-specific features. The reason is that there commonly exist multiple factors that contribute to domain-specific features. Taking the identification of "elephants" and "cats" as an example, the hypotheses "elephants tend to be found in grasslands", and "elephants tend to have wrinkled skin" are both beneficial for classification. When we deploy classifiers to the real-world, these hypotheses are domain-specific and superficial, and might result in severe performance degradation. For images, there are several factors typically correlated to domain-specific features, such as the background contexts [1], the texture of the objects [19], and high-frequency patterns that are almost invisible to the human eye [55]. That motivates us to design a biased-branch that identifies more domain-specific features with a multi-head cooperated classifier $g_{bias} : \mathbb{R}^D \rightarrow \mathbb{R}^{K\times M}$. Specifically, we apply the cooperation cross-entropy loss to learn this branch as:

$$\mathcal{L}_{C-CE} = \mathbb{E}_{x,y} \sum_{k=1}^{K} -\mathbb{1}_{[k=y]} \log \frac{\exp(\max(v_k(x)))}{\sum_{j=1}^{K} \exp(\max(v_j(x)))}, \tag{1}$$

where $v_k(x) = g_{bias}(f(x))[k,:] \in \mathbb{R}^M$ denotes the logits of multi-head classifier for the k-th category of sample x, and M is the number of classification heads. Note that we do not enforce all heads of the biased-branch classifier to correctly predict each sample. Instead, we only need one of them to accurately identify it. That is, all heads are encouraged to cooperate with each other for classification. The spirit behind is that domain-specific features do not represent the truly domain-generalized semantics. Thereby, for a particular-type domain-specific features, they are not necessarily present in all samples. Since the *max* function is not differentiable in Eq. (1), we approximate this function with the *log-sum-exp* during our implementation.

In general, the sweet spot for M is set within the range from 1 to $D//K - 1$. Its value depends on the dimension and factors introduced domain-specific features. In our implementation, we perform cross validation to choose a good value for M, but it is worthy to note that the performance is relatively stable with respect to this choice (see more discussions in Sec. 4.5).

3.3. Learning to Debias

Based on the proposed biased-branch, we have an indicator to those domain-specific features. A follow-up question is how to suppress those domain-specific features in favor of focusing more on those desired domain-generalized features. Here, we introduce another general-branch classifier $g_{gen} : \mathbb{R}^D \rightarrow \mathbb{R}^K$ to capture those domain-generalized features. Let $W_{bias} \in \mathbb{R}^{K\times M\times D}$ and $W_{gen} \in \mathbb{R}^{K\times D}$ be weights of the multi-head biased classifier and the domain-general classifier, respectively. An intuitive solution is to enforce orthogonality between W_{bias} and W_{gen} in Eq. (3) during learning the classifier g_{gen} in Eq. (2):

$$\mathcal{L}_{CE} = \mathbb{E}_{x,y} \sum_{k=1}^{K} -\mathbb{1}_{[k=y]} \log \frac{\exp(u_k(x))}{\sum_{j=1}^{K} \exp(u_j(x))}, \tag{2}$$

$$\mathcal{L}_{Reg} = \frac{1}{K} \sum_{k=1}^{K} \left\| W_{bias}[k,:]\, W_{gen}[k,:]^T \right\|_F^2 . \tag{3}$$

Here $u_k(x) = g_{gen}(f(x))[k] \in \mathbb{R}$ represents the logit of classifier g_{gen} for the k-th category of input sample x. However, if we optimize the whole network (including the feature extractor f, biased-branch classifier g_{bias}, and general-branch classifier g_{gen}) simultaneously at the beginning, there is no guarantee that the classifier g_{gen} will pay more attention to those domain-general features. To address this issue, we introduce a two-stage learning mechanism to enable the interaction between the two branches. Technically, in the first stage, we only introduce Eq. (1) and Eq. (3) to optimize the network, encouraging the biased-branch classifier to learn those domain-specific features and expecting the weight of general-branch classifier W_{gen} to evade the territory of domain-specific features. Then, in the second stage, we apply Eq. (1), Eq. (3) and Eq. (2) together

Algorithm 1 MAD Pseudocode, PyTorch-like

```
# f: feature extractor
# cls_g1: the biased-branch multi-head classifier
# cls_g2: the general-branch classifier
# pro, T: training progress, second stage thresh

# During inference, we only utilize f, and cls_g2

for x, y in loader: # load a minibatch x, y
   z = f(x) # (N, D)
   c1 = cls_g1(z) # (N, K, M)
   c1 = logsumexp(c1, dim=-1) # (N, K)
   l_bias = CrossEntropyLoss(c1, y)

   c2 = cls_g2(z) # (N, K)
   l_gen = CrossEntropyLoss(c2, y)

   g1_w = cls_g1.weight # (K, M, D)
   g2_w = cls_g2.weight # (K, D)
   l_reg = Reg(g1_w, g2_w)

   if pro < T:
      L = l_bias + l_reg # stage 1
   else:
      L = l_bias + l_reg + l_gen # stage 2

   L.backward() # back-propagate
   update(f, cls_g1, cls_g2) # SGD update

def Reg(w1, w2): # orthogonality regulation
   w1 = normalize(w1, dim=-1) # l2-normalize
   w2 = normalize(w2, dim=-1) # l2-normalize
   reg = einsum("kmd,kd->km", w1, w2)
   return mean(sum(reg ** 2, dim=-1))
```

to optimize the entire network. Accordingly, the overall optimization objective is:

$$\min_{f,g_{bias},g_{gen}} \mathcal{L}_{C-CE} + \mathcal{L}_{Reg} + \mathbb{1}_{[pro\geq T]} \cdot \mathcal{L}_{CE}, \qquad (4)$$

where all loss terms are equally weighted, pro denotes the overall training progress, T is a hyper-parameter that determines when to trigger the second stage learning. In general, the choice of T depends on the training dataset size and task difficulty. In our implementation, for recognition task, we typically set $T = 3$ epochs (50 epochs in total). As for semantic segmentation, we set $T = 6\%$ of the iterations in total. Algorithm. 1 presents the Pseudo-code of our MAD.

4. Experiments

We evaluate the effectiveness of MAD for single domain generalization (single-DG) via various empirical evidences on a series of tasks, including recognition on images, point clouds, texts, and semantic segmentation on images. Here we include several single-DG methods as baselines for performance comparison: (1) ERM [49] directly applies the vanilla strategy to train source model. (2) AugMix [24] utilizes stochastic and diverse augmentations, and a formation to mix multiple augmented images to generate diverse samples. (3) pAdaIN [34] swaps feature statistics between the samples applied with a random permutation of mini-batch, (4) Mixstyle [65] adopts linear interpolation on feature statistics of two instances to generate synthesized samples. (5) DSU [30] characterizes the feature statistics as uncertain distribution to model domain shift. (6) ACVC [15] introduces more severe image augmentations, including image corruptions and Fourier transform. Recall that our MAD is able to directly strengthen the capability of classifier to identify domain-specific features, and meanwhile emphasize the learning of domain-generalized features. Therefore, MAD can be seamlessly incorporated into these methods to further boost performances. Note that MAD discards the additional biased-branch and only employs the feature extractor plus general-branch classifier at inference. That is, when plugging MAD into existing methods, there is no increase in computational cost.

4.1. Single-DG on Image Recognition

Setup and Implementation Details: We validate the proposed method on two image datasets: **PACS** [29], a widely-used benchmark for domain generalization with four domains: Photo (P), Art Painting (A), Cartoon (C), and Sketch (S). **VLCS** [46], another commonly adopted benchmark for domain generalization with four different domains: VOC2007 (V), LabelMe (L), Caltech101 (C), SUN09 (S). In our implementation, we adopt the ResNet-18 [23] pretrained on ImageNet [16] as backbone. We apply the SGD optimizer with momentum 0.9. The batch size is set to 64. We set the learning rate to 2e-3/1e-3 for PACS/VLCS. Experiments are conducted on a Tesla P40 GPU with PyTorch-1.5. Following [22], we split the training domain into training and validation subsets, and select the best-performing model on validation set to report the OOD performances.

Experiment Results: We first conduct experiments on PACS, shown in Table 1. The main domain shift in this dataset is derived from style differences, and most data augmentation methods manifest higher performances than ERM baseline. Though these methods have achieved good performances, our MAD still manages to further improve their performance consistently. For example, MAD boosts up the overall accuracy of ACVC from 63.61% to 65.87%. Table 2 further summarizes the performance comparison on VLCS. The domain shift of this dataset mainly comes from background and view point changes. The scenes in VLCS vary from urban to rural, and the viewpoint tends to favor the side view or non-classical view. As a result, existing data augmentation methods which mainly introduce diverse styles obtain relatively smaller performance gains on VLCS dataset compared to those on PACS. Even in this case, MAD significantly improves the overall accuracy of ERM from 59.56% to 62.95%. Especially when taking “LabelMe” as source domain, our MAD leads to near 10% improvement in average accuracy. Similar to the observations on PACS, the consistent performance improvements are attained when integrating existing data augmentation approaches with MAD. In particular, MAD increases the accuracy of ACVC from 61.25% to 63.82%.

Table 1. Single-domain generalization classification accuracies (%) on PACS dataset with ResNet-18 as backbone. Here, P, A, C, and S denote the source domains. We train the model on one source domain, and evaluate them on the rest domains.

Methods	Venue	P	A	C	S	Avg
ERM		33.65	65.38	64.20	34.15	49.34
ERM w/ MAD		32.32	66.47	69.80	34.54	**50.78**
Augmix [24]	ICLR 19	38.30	66.54	70.16	52.48	56.87
Augmixw/ MAD		36.19	68.04	73.11	54.44	**57.94**
pAdaIn [34]	CVPR 21	33.66	64.96	65.24	32.04	48.98
pAdaIn w/ MAD		34.66	65.64	70.10	42.85	**53.31**
Mixstyle [65]	ICLR 21	37.44	67.60	70.38	34.57	52.50
Mixstyle w/ MAD		41.57	69.88	71.61	41.58	**56.16**
ACVC [15]	CVPR 22	48.05	73.68	77.39	55.30	63.61
ACVC w/ MAD		52.95	75.51	77.25	57.75	**65.87**
DSU [30]	ICLR 22	42.10	71.54	74.51	47.75	58.97
DSU w/ MAD		44.15	72.41	74.47	49.60	**60.16**

Table 2. Single-domain generalization classification accuracies (%) on VLCS dataset with ResNet-18 as backbone. Here, V, L, C, and S denote the source domains. We train the model on one source domain, and evaluate them on the rest domains.

Methods	Venue	V	L	C	S	Avg
ERM		76.72	58.86	44.95	57.71	59.56
ERM w/ MAD		76.21	67.97	46.55	61.04	**62.95**
Augmix [24]	ICLR 19	75.25	59.52	45.90	57.43	59.53
Augmixw/ MAD		76.57	65.60	44.35	59.47	**61.50**
pAdaIn [34]	CVPR 21	76.03	65.21	43.17	57.94	60.59
pAdaIn w/ MAD		76.57	68.90	42.92	63.91	**63.08**
Mixstyle [65]	ICLR 21	75.73	61.29	44.66	56.57	59.56
Mixstyle w/ MAD		75.00	66.17	43.61	62.01	**61.70**
ACVC [15]	CVPR 22	76.15	61.23	47.43	60.18	61.25
ACVC w/ MAD		76.15	69.36	48.04	61.74	**63.82**
DSU [30]	ICLR 22	76.93	69.20	46.54	58.36	62.76
DSU w/ MAD		76.99	70.85	44.78	62.23	**63.71**

4.2. Single-DG on Point Cloud Recognition

Setup and Implementation Details: Different from 2D vision, 3D vision has various modalities to represent data, such as voxel grid, 3D mesh and point cloud. Among them, point cloud is the most straightforward and representative modality, which consists of a set of points with 3D coordinates. To verify the generality of MAD, we conduct experiments on the 3D point cloud domain adaptation dataset **PointDA-10** [39], which consists of three domains: ShapeNet (SH), ScanNet (SC), and ModelNet (M). In our implementation, we adopt the PointNet [39] as backbone, and apply the SGD optimizer with momentum 0.9. The batch size is set to 64. We set the learning rate to 1e-3. Experiments are executed on a Tesla P40 GPU with PyTorch-1.5. For model selection, similar to the experiments on images, we split the training domain into training and validation subsets, and choose the model with maximal accuracy on validation set to report the OOD performance.

Table 3. Single-domain generalization classification accuracies (%) on PointDA-10 dataset with PointNet as backbone. Here, SH, SC, and M denote the source domains. We train the model on one source domain, and evaluate them on the rest domains.

Methods	Venue	SH	SC	M	Avg
ERM	-	25.69	45.09	32.94	34.57
ERM w/ MAD		31.11	48.07	34.69	**37.91**
Mixstyle [65]	ICLR 21	27.18	46.25	27.93	33.78
Mixstyle w/ MAD		29.89	51.01	33.57	**38.16**
DSU [30]	ICLR 22	25.74	43.53	31.61	33.63
DSU w/ MAD		28.92	47.69	32.72	**36.45**

Experiment Results: Table 3 lists the performance comparison on PointDA-10. An observation is that the existing data augmentation methods on 2D images do not work well on 3D point clouds. The representative methods, e.g., Mixstyle and DSU, are even inferior to ERM. We speculate that this may be the results of the different types of domain shifts, which typically lie in geometric differences in point clouds rather than texture and style differences in 2D images. Moreover, there is no one-to-one correspondence and order between points, making it difficult to directly generate new point clouds by interpolating two point clouds. This somewhat reveals the weakness of data augmentation, when generalizing to different modalities. MAD, in comparison, benefits from decoupling domain-specific features and domain-generalized features, and constantly enhances these methods. In particular, MAD improves the overall accuracy of ERM/Mixstyle/DSU from 34.57%/33.78%/33.63% to 37.91%/38.16%/36.45%. The results basically indicate the advantage of MAD across different modalities.

4.3. Single-DG on Text Classification

Setup and Implementation Details: In addition to 2D images, and 3D point clouds, we further conduct experiments on cross-domain text classification. We choose the **Amazon Reviews** [3] as the benchmark, which contains four different domains on product review, including DVDs (D), Kitchen appliance (K), Electronics (E), and Books (B). The dataset has already been pre-processed into a bag of features (unigrams and bigrams), losing all word order information. Following [8, 9], we take the 5,000 most frequent features and represent each review as a 5,000-dimentional feature vector. Following [8, 9], we employ an MLP as feature extractor. We apply the SGD optimizer with momentum 0.9. The batch size is set to 64. We set the learning rate to 1e-3. Experiments are conducted on a Tesla P40 GPU with PyTorch-1.5. We adopt the same model selection strategy as in image and point cloud recognition.

Experiment Results: The results shown in Table 4 clearly

Table 4. Single-domain generalization classification accuracies (%) on Amazon-Review dataset with an MLP as backbone. Here, D, E, K, and B denote the source domains. We train the model on one source domain, and evaluate them on the rest domains.

Methods	Venue	D	E	K	B	Avg
ERM	-	74.17	73.17	73.67	71.58	73.15
ERM w/ MAD		76.08	74.33	73.33	74.67	**74.60**
Mixup [61]	ICLR 18	74.83	72.17	73.58	72.67	73.31
Mixup w/ MAD		75.33	73.58	74.33	73.75	**74.25**
Mixstyle [65]	ICLR 21	74.75	73.17	74.33	72.33	73.65
Mixstyle w/ MAD		75.17	72.75	75.00	75.25	**74.54**
DSU [30]	ICLR 22	75.00	73.45	75.25	73.08	74.20
DSU w/ MAD		76.42	74.33	76.50	75.17	**75.60**

Table 5. Single-domain generalization on semantic segmentation (GTA-5 → Cityscapes).

Methods	Venue	mIOU(%)	mACC(%)
ERM	-	37.0	51.5
pAdaIN [34]	CVPR 21	38.3	52.1
Mixstyle [65]	ICLR 21	40.3	53.8
DSU [30]	ICLR 22	42.3	54.7
ERM w/ MAD	-	**38.9**	**52.2**
DSU w/ MAD	-	**43.8**	**57.2**

verify the effectiveness of MAD in comparison to the existing methods. Similar to the observations on 2D images and 3D point clouds, MAD also exhibits performance improvement to existing approaches on text modality. For example, MAD boosts up the accuracy of ERM on Books domain by 3.09%, and leads to 0.94%, 0.89%, and 1.40% gain in overall accuracy to Mixup, Mixstyle, and DSU, respectively. The improvements empirically prove the impact of MAD on text modality.

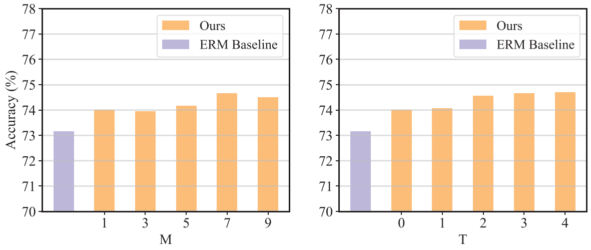
4.4. Single-DG on Semantic Segmentation

Setup and Implementation Details: The aforementioned experiments mainly focus on the single-DG recognition of 1D texts, 2D images and 3D point clouds. In this section, we experiment with 2D images segmentation. As a fundamental ability for autonomous driving, semantic segmentation models often encounter severe performance degeneration due to scenarios change. Here, we conduct experiments on GTA-5 [42] → Cityscape [13] datasets, the most widely-used benchmark on semantic segmentation domain adaptation. The experiments are based on FADA released codes [54], using DeepLab-V2 [7] segmentation network with ResNet-101 [23] as backbone. We apply the SGD optimizer with momentum 0.9. The batch size is set to 8. We set the learning rate to 5e-4. Experiments are implemented on 4 Tesla P40 GPUs with PyTorch-1.5. Mean Intersection over Union (mIOU) and mean Accuracy (mAcc) for all objects categories are adopted as evaluation metric.

Experiment Results: As a pixel-level classification task, semantic segmentation is much harder than image-level recognition. Table 5 details the results, demonstrating the superiority of MAD against baselines. Specifically, MAD contributes an mIOU increase of 1.9% and 1.5% to ERM and DSU, respectively. The results again verify the merit of MAD on semantic segmentation on 2D images.

Table 6. **Ablation**. Results of the vanilla ERM, ERM w/ MAD (one-stage), ERM w/ MAD (single-head), and ERM w/ MAD. The experiments are conducted on single-domain generalization scenarios of 1D Texts (Amazon Review dataset), 2D Images (VLCS dataset), and 3D Point Clouds (PointDA-10 dataset).

Methods	1D Texts	2D Images	3D Points	Avg
ERM	73.15	59.56	34.57	55.76
MAD (one-stage)	74.00	60.41	35.67	56.69
MAD (single-head)	74.31	60.49	36.61	57.14
MAD	**74.60**	**62.95**	**37.91**	**58.49**



(a) sensitivity to M (T = 3)

(b) sensitivity to T (M = 7)

Figure 3. Hyper-parameter sensitivity analysis on Single-DG text classification. M denotes the number of the biased-branch classifiers and T is the training threshold for the second stage.

4.5. Experiments Analysis

Ablation Study: To examine the contribution of different components within MAD, we first conduct extensive ablation studies on texts, images, and point clouds recognition. Table 6 summarizes the results. Here, *MAD (one-stage)* refers to a degraded version of MAD without two-stage learning mechanism. That is, we optimize the biased-branch g_{bias} and the general-branch g_{gen} simultaneously. *MAD (single-head)* indicates that we only capitalize on a single-head classifier in the biased-branch to capture those domain-specific features. As shown in Table 6, both the multi-head classifier design and the two-stage learning mechanism are effective. The two components complement to each other and both manage the general-branch classifier g_{gen} to focus more on those domain-generalized features.

Hyper-parameter Sensitivity: Next, we study the hyper-parameter sensitivity of M and T on text classification task. M is the number of the biased-branch classifiers, and T denotes the second-stage training threshold. As shown in Figure 3, the accuracies are relatively stable when each hyper-parameter varies. In our implementation, we set T to 3. Since M depends on the factors of the introduced domain-

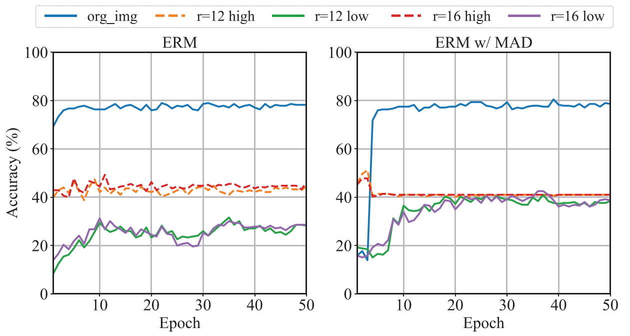


Figure 4. Comparisons of ERM and ERM w/ MAD training curves on low-frequency component (LFC) and high-frequency component (HFC). Experiments are conducted in the “LabelMe” domain of the VLCS benchmark. All curves in this figure are from validation samples of “LabelMe” domain.

specific features, its value differs for different datasets. We set M to 3 for PointDA-10 and GTA-5, 5 for VLCS, and 7 for PACS and Amazon Review.

Low-frequency Component vs High-frequency Component: As pointed out in [55], the low-frequency component (LFC) is much more generalizable than high-frequency component (HFC), i.e., LFC typically represents those domain-generalized (semantic) features, and HFC denotes those domain-specific (superficial) features. Here, we conduct experiments in the “LabelMe” domain of the VLCS benchmark to verify whether MAD encourages classifier to pay more attention to those domain-generalized features, i.e., the LFC. Specifically, for each instance in the validation subset, we decompose the data into LFC and HFC w.r.t different radius thresholds r via applying Fourier transform and inverse Fourier transform. Then, we train the vanilla ERM classifier, the ERM classifier equipped with MAD separately, and evaluate them on LFC and HFC. Figure 4 depicts the results, where $r = 12/16\ low$ (solid line) denotes the LFC and $r = 12/16\ high$ (dashed line) denotes the HFC. As shown in this figure, ERM w/ MAD performs much better on LFC than vanilla ERM, with an accuracy improvement of nearly 10%. The results confirm the effectiveness of MAD in improving single-domain generalization, and MAD indeed encourages classifiers to pay more attention to those domain-generalized features (LFC).

4.6. Visualization

In addition to quantitative performance comparisons, we further present some qualitative illustrative results. Figure 5 (a) first visualizes the confusion matrix on PACS benchmark. The classification model is trained on “Cartoon” domain and evaluated on unseen domain “Sketch”. The results show that ERM w/ MAD is less confusing for most categories when testing in unseen domains compared to vanilla ERM. Then, Figure 5 (b) illustrates an example for semantic segmentation. The visualization demonstrates that MAD can enhance the ERM baseline to achieve more precise segmentation results under domain shift, especially for the driveable areas.

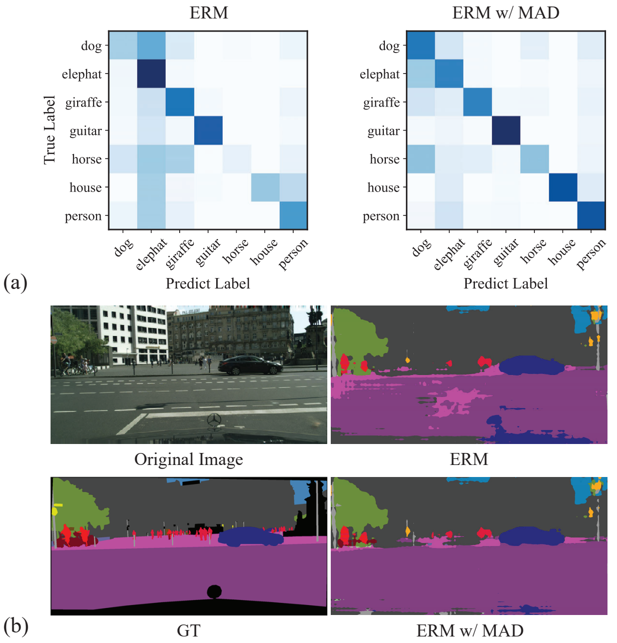


Figure 5. **Visualization.** (a) Confusion matrix on PACS benchmark (“Carton”→ “Sketch”). (b) Semantic segmentation illustration on unseen domain Cityscapes with model trained on GTA-5.

5. Conclusion

In this paper, we delve into the single domain generalization (single-DG) problem. Different from existing methods that introduce modality-specific data augmentation techniques, we propose a general and versatile modality-agnostic debiasing (MAD) framework for single-DG. MAD starts from the viewpoint of directly strengthening the capability of classifier for identifying domain-specific (superficial) features, and meanwhile emphasizing the learning of domain-generalized (semantic) features. Technically, we have devised a novel two-branch classifier, where a biased-branch is responsible for identifying those superficial features, while the general-branch is encouraged to focus more on those semantic features. MAD is appealing in view that it can be seamlessly incorporated into existing methods to further boost up performances. We have evaluated the effectiveness and superiority of MAD for single-DG via various empirical evidences on a series of tasks, including recognition on 1D texts, 2D images, 3D point clouds, and semantic segmentation on 2D images. In all tasks, MAD can facilitate the state-of-the-art methods to achieve better performance without bells and whistles.

Acknowledgment: This work is supported by Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), ZJ Lab, and Shanghai Center for Brain Science and Brain-Inspired Technology, and the Shanghai Rising Star Program (No.21QC1400900).

References

[1] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. 4

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

[3] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, 2006. 6

[4] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, 2019. 2

[5] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *NeurIPS*, 2021. 2

[6] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*, 2020. 4

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 2017. 7

[8] Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. Multi-source cross-lingual model transfer: Learning what to share. In *ACL*, 2019. 6

[9] Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *NAACL*, 2018. 6

[10] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *ECCV*, 2020. 2

[11] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018. 2

[12] Yang Chen, Yu Wang, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. A style and semantic memory mechanism for domain generalization. In *ICCV*, 2021. 4

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 7

[14] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE TPAMI*, 39(9):1853–1865, 2016. 2

[15] Ilke Cugu, Massimiliano Mancini, Yanbei Chen, and Zeynep Akata. Attention consistency on visual corruptions for single-source domain generalization. In *CVPR Workshop*, 2022. 3, 5, 6

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[17] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1):110–125, 2002. 2

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. 1, 2

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 1, 4

[20] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE TPAMI*, 2016. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 2

[22] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. 5

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[24] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 5, 6

[25] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 2

[26] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *WACV*, 2020. 2

[27] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis. In *UAI*. PMLR, 2020. 2

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[29] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 5

[30] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling-Yu Duan. Uncertainty modeling for out-of-distribution generalization. In *ICLR*, 2022. 1, 2, 3, 5, 6, 7

[31] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018. 1, 2

[32] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013. 1, 2

[33] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 2

[34] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: Reducing the bias towards global statistics in image classification. In *CVPR*, 2021. 5, 6, 7

[35] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *CVPR*, 2020. 2

[36] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, 2019. 1

[37] Xi Peng, Fengchun Qiao, and Long Zhao. Out-of-domain generalization from a single source: An uncertainty quantification approach. *IEEE TPAMI*, 2022. 1, 2, 3

[38] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*, 2020. 4

[39] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In *NeurIPS*, 2019. 2, 6

[40] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *ECCV*, 2022. 1, 2

[41] Sanqing Qu, Tianpei Zou, Florian Roehrbein, Cewu Lu, Guang Chen, Dacheng Tao, and Changjun Jiang. Upcycling models under domain and category shift. In *CVPR*, 2023. 1, 2

[42] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 7

[43] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, 2020. 2

[44] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *NeurIPS*, 2020. 3

[45] Yichun Shi, Xiang Yu, Kihyuk Sohn, Manmohan Chandraker, and Anil K Jain. Towards universal representation learning for deep face recognition. In *CVPR*, 2020. 2

[46] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 5

[47] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *WACV*, 2021. 2

[48] Vladimir Vapnik. Principles of risk minimization for learning theory. In *NeurIPS*, 1991. 1, 3

[49] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999. 5

[50] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *ICCV*, 2019. 2

[51] Chaoqun Wan, Xu Shen, Yonggang Zhang, Zhiheng Yin, Xinmei Tian, Feng Gao, Jianqiang Huang, and Xian-Sheng Hua. Meta convolutional neural networks for single domain generalization. In *CVPR*, 2022. 3

[52] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, 2020. 4

[53] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. In *ICLR*, 2019. 1, 3

[54] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. 7

[55] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 2020. 1, 4, 8

[56] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE TKDE*, 2022. 1, 2

[57] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[58] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021. 1, 2, 3

[59] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *ICLR*, 2021. 2

[60] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015. 2

[61] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 7

[62] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P Xing. Towards principled disentanglement for domain generalization. In *CVPR*, 2022. 1

[63] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 1

[64] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE TIP*, 2021. 2

[65] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 1, 2, 5, 6, 7