BoosterNet: Improving Domain Generalization of Deep Neural Nets using Culpability-Ranked Features

Nourhan Bayasi
University of British Columbia
nourhanb@ece.ubc.ca

Ghassan Hamarneh
Simon Fraser University
hamarneh@sfu.ca

Rafeef Garbi
University of British Columbia
rafeef@ece.ubc.ca

Abstract

Deep learning (DL) models trained to minimize empirical risk on a single domain often fail to generalize when applied to other domains. Model failures due to poor generalizability are quite common in practice and may prove quite perilous in mission-critical applications, e.g., diagnostic imaging where real-world data often exhibits pronounced variability. Such limitations have led to increased interest in domain generalization (DG) approaches that improve the ability of models learned from a single or multiple source domains to generalize to out-of-distribution (OOD) test domains. In this work, we propose BoosterNet, a lean add-on network that can be simply appended to any arbitrary core network to improve its generalization capability without requiring any changes in its architecture or training procedure. Specifically, using a novel measure of feature culpability, BoosterNet is trained episodically on the most and least culpable data features extracted from critical units in the core network based on their contribution towards class-specific prediction errors, which have shown to improve generalization. At inference time, corresponding test image features are extracted from the closest class-specific units, determined by smart gating via a Siamese network, and fed to BoosterNet for improved generalization. We evaluate the performance of BoosterNet within two very different classification problems, digits and skin lesions, and demonstrate a marked improvement in model generalization to OOD test domains compared to SOTA.

1. Introduction

The remarkable advances in deep learning (DL) models rendered deep neural networks (DNNs) ubiquitous in various fields, particularly in computer vision including safety-critical applications such as medical image analysis [21, 32, 60]. Despite their relative success when applied to new data in certain applications, practical deployment of DNN based solutions remains very risky with one of the main concerns being vulnerability to domain shifts which leads to poor generalizability to out-of-distribution (OOD) data. Such limitations not only impair model performance but can result in serious unacceptable failures when test data is drawn from a different distribution than that of the training data [16,43,58]. This unpredictable performance degradation on real life data continues to hamper reliable practical deployment such as in healthcare.

Recognizing this serious problem, much research has recently focused on improving model generalizability. In *unsupervised domain adaptation (UDA)*, the aim is to transfer the knowledge of a label-rich training domain to unlabeled test domains with the same classes as those of the training data [31, 36, 49, 63, 66]. However, UDA methods have limited value due to the requirement of accessing some of the test data which may not be available in advance. In *Domain generalization (DG)* approaches, the aim is to instead utilize a single or multiple source domains' information to better generalize to OOD domains without requiring any access to the test data. The field of DG is quite rich with a spectrum of techniques ranging from domain alignment [28, 46] to data augmentation [51, 57, 65], meta-learning [7, 34] and ensemble learning [45, 55]. However, despite significant performance improvements, most DG approaches still suffer from common drawbacks. First, they typically require training data from multiple domains, which can be rather challenging, costly and even infeasible, e.g., due to privacy issues in medical data applications. Second, they often require restructuring or changes in the network architecture or learning strategy to achieve the desired performance [8,9,19,30]. For end users who are not seasoned data scientists, e.g., dermatologists trying to classify skin lesions with minimal or no training in DL, such amendments are impractical.

In this work, we propose a single-source DG framework for improving generalizability of an arbitrary off-the-shelf DNN (core network $\mathcal{A}$) by learning from its mistakes. We argue that BoosterNet ameliorates *shortcut learning and feature suppression*, a problem that has only recently gained more attention [12, 18, 44], where in the presence of multiple predictive input features, a model tends to only use a subset and ignores the other features often leading to 'shortcut' decision rules that might perform well on training data but would harm generalization ability and lead to poor robustness to data shifts. To combat shortcut learning and improve generalization capabilities, **BoosterNet** comprises a lean add-on network that is encouraged to learn, through episodic training, from the most culpable features in the core network $\mathcal{A}$ most associated with erroneous prediction (hereafter referred to as *confusion features*). To balance the learning process, BoosterNet is trained to also retain focus on the most predictive 'trivial' characteristics of the data, namely by training on the least culpable features as well (hereafter referred to as *discriminant features*). A high level overview of our DG framework is illustrated in Figure 1 (training and inference details are illustrated in Figure 2). Using our proposed culpability score, the confusion and discriminant features are extracted from class-specific units (filters/neurons) in network $\mathcal{A}$ being associated with the highest and lowest *culpability* in erroneous prediction in a specific class, respectively. At inference time, BoosterNet processes a test image by extracting confusion and discriminant features corresponding to the closest class-units as determined by a smart gating mechanism based on a Siamese network. Our extensive experiments illustrate that our method outperforms state-of-the-art (SOTA) in single domain generalization on classification benchmark datasets including digits and medical skin images.

2. Related Work

Domain generalization (DG) has been intensively studied in recent years. Early DG approaches mainly focused on data preparation and augmentation by creating new diverse training data samples to encourage models to learn general representations that may better support generalization [25,42,52,59,64]. While such methods only increased the source capacity, an exception was proposed in [41, 61] where both input and label spaces were augmented. Differently, [57] developed randomized convolutions as a data augmentation technique to stimulate an infinite number of new domains with similar global shapes but random local texture to improve model generalization.

In the context of DG methods focusing on representation learning, the goal is to learn domain-invariant features that are general and transferable to different domains. Muandet et al. [13] proposed a kernel-based method to obtain domain invariant features. [37, 58] proposed a cross-domain contrastive semantic alignment (CCSA) loss that encourages intra-class similarity and inter-class difference across all domains. Other methods explicitly minimized feature distribution divergence across domains by minimizing either the maximum mean discrepancy (MMD) [54], second order correlation [40] or moment matching [39]. Learning domain-invariant features has also been performed via domain-adversarial learning [28,33,46] as well as by modifying the core network architecture [9,30,38,45].

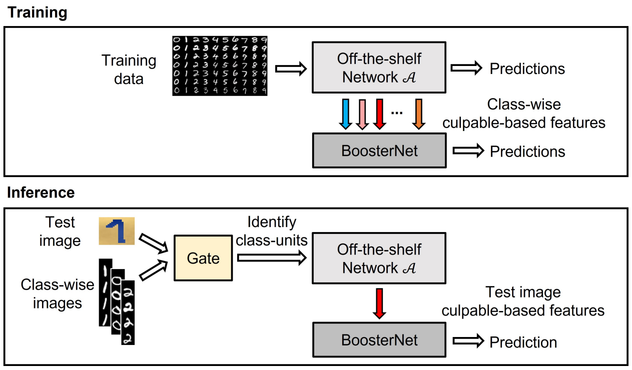
Training
Training data
Off-the-shelf Network $\mathcal{A}$
Predictions
Class-wise culpable-based features
BoosterNet
Predictions
Inference
Test image
Class-wise images
Gate
Identify class-units
Off-the-shelf Network $\mathcal{A}$
Test image culpable-based features
BoosterNet
Prediction

Figure 1. Overview of BoosterNet. (Top) Training: BoosterNet is trained on 'confusion' and 'discriminant' features extracted from class-specific units in network $\mathcal{A}$ with the highest and lowest culpability scores, respectively. (Bottom) Inference: Using a Siamese network as a smart gate, the closest class of the test image is identified based on the shortest Euclidean distance and the corresponding test image features are extracted from network $\mathcal{A}$ and fed to BoosterNet for prediction.

Recently, with the rise of model-agnostic meta learning (MAML) [10], meta-learning methods have become popular [14, 20, 26, 35, 47]. The idea is to simulate a virtual meta-task by adopting an episodic training paradigm, i.e., splitting the available training domains into meta-train and meta-test at each iteration to simulate the domain shift. Li et al. [27] designed an episodic training procedure that decomposes a deep network into feature extractor and classifier components and then trains each component by simulating it interacting with a partner which is badly tuned for the current domain. In this manner, both feature extractor and classifier components become robust and generalizable. Dou et al. [6] introduced two complementary meta-losses which explicitly regularize the semantic structure of the feature space via a model-agnostic episodic learning procedure. [24] proposed a new episodic learning framework where the meta-test data is generated by interpolating all source domains to enhance the variety of the meta-task simulation. While contributing positively to address the generalization problem, all the above methods rely on the availability of a number of training domains to avoid overfitting.

Most recently, the Fourier Transform has become a hot area of research for domain generalization [56, 62]. The main assumption is that Fourier phase information generally contains high-level semantics that are not easily affected by domain shifts. Thus, by learning from phase information, the model may better extract the semantic concepts from different image data that could prove robust to domain shifts.

BoosterNet. The existing DG methods require changes in network architecture or optimization, which could be challenging in some applications or for some users. Our proposed BoosterNet can be easily coupled with any core network without requiring any of the changes. BoosterNet can be viewed as a harmonious combination of data preparation and episodic training-based DG approaches since we leverage the training data to extract culpable-based features and use episodic training to simulate domain shift and improve generalization. To summarize, we make the following contributions:

- We propose using the concept of feature culpability to improve domain generalization from a single-source domain. Our culpability score measures the contribution of each network unit towards erroneous class-wise predictions.
- We propose BoosterNet, a simple network that acts as an add-on to existing DNN networks to boost generalizability by leveraging information from the least and most culpable features to learn more generalizable predictive features.
- We conduct extensive experiments on two applications with multi-domain datasets including digits and skin lesion classification with 4 and 5 domains, respectively, to validate the effectiveness of our framework, and demonstrate superior performance over SOTA in improving domain generalization.

3. Methods

We first describe our problem setting and overall framework design. BoosterNet is a lean add-on network that can be coupled with an arbitrary DNN (network $\mathcal{A}$) to improve generalization without changing its architecture or optimization. Given a single-source training domain $\mathcal{S}$, network $\mathcal{A}$ is trained through standard supervised learning. BoosterNet learns from culpability-ranked features (namely, confusion and discriminant features) extracted from network $\mathcal{A}$ to improve generalization on OOD target domains $\{\mathcal{T}_1, \mathcal{T}_2, \cdots\} \sim p(\mathcal{T})$. BoosterNet training and inference are summarised in Figure 2.

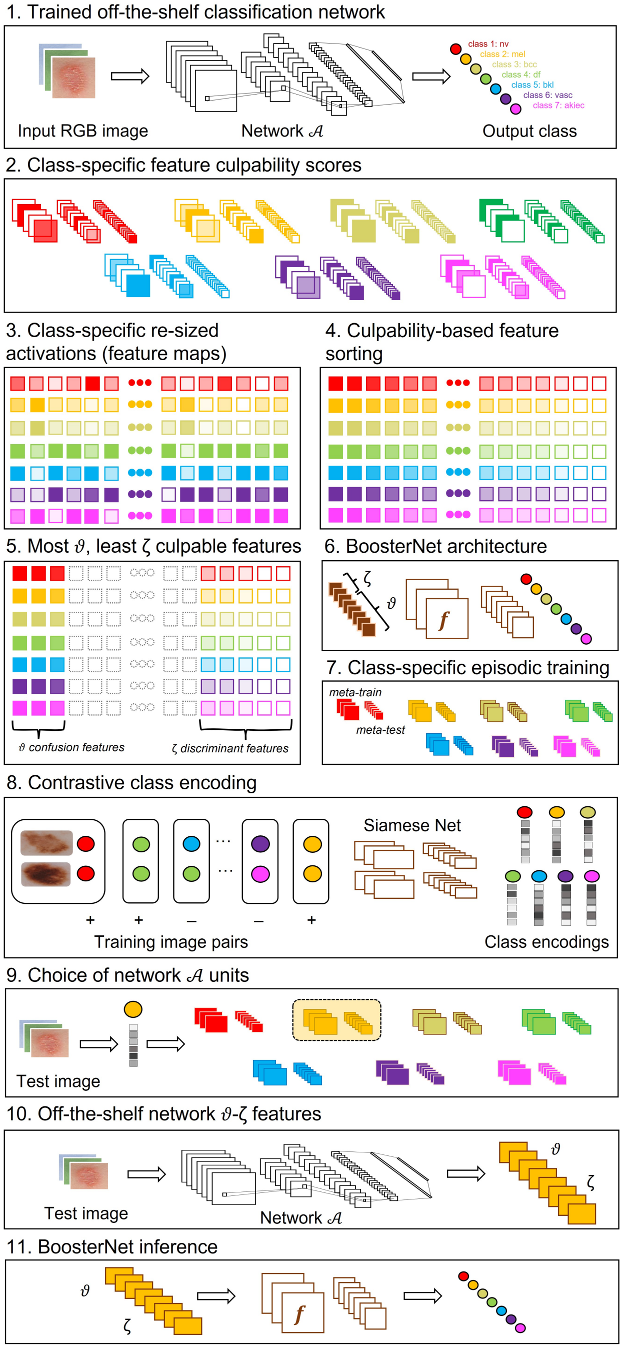
1. Trained off-the-shelf classification network
Input RGB image
Network $\mathcal{A}$
Output class
class 1: nv
class 2: mel
class 3: bcc
class 4: df
class 5: bkl
class 6: vasc
class 7: akiec
2. Class-specific feature culpability scores
3. Class-specific re-sized activations (feature maps)
4. Culpability-based feature sorting
5. Most ϑ, least ζ culpable features
ϑ confusion features
ζ discriminant features
6. BoosterNet architecture
ζ
ϑ
f
7. Class-specific episodic training
meta-train
meta-test
8. Contrastive class encoding
Siamese Net
+
+
–
–
+
Training image pairs
Class encodings
9. Choice of network $\mathcal{A}$ units
Test image
10. Off-the-shelf network ϑ-ζ features
Test image
Network $\mathcal{A}$
ϑ
ζ
11. BoosterNet inference
ϑ
ζ
f

Figure 2. Details of BoosterNet training stage (steps 1 to 7) and inference stage (steps 8 to 11) added onto a traditional network $\mathcal{A}$. In this example, our application case study is skin lesion classification, where the training dataset used is HAM [50] with RGB input images categorized into seven skin lesion classes.

3.1. Culpability Scoring

Our *culpability score* $[C]_{m,n}$ quantifies the contribution of each unit n in a network towards erroneous prediction for each class m in the training dataset $\mathcal{S}$ [2]. To calculate the scores, a pre-trained off-the-shelf classification network $\mathcal{A}$, with parameters θ and n units (filters/neurons), predicts the class output y given an input image x through standard supervised learning, e.g., cross-entropy loss (Figure 2 (step 1)). After training, we group the validation data $\{x, y\}$ into two groups per class m: $\mathcal{R}_m$ is the set of (x, y) pairs with ground truth class m and predicted class m (i.e., $\underline{R}$ightly classified images); $\mathcal{W}_m$ is the set of (x, y) pairs with ground truth class m but predicted class other than m (i.e., $\underline{W}$rongly classified images). To compute the culpability score for each unit n, we analyze the rectified activations a_i^n of the input data sample i across $\mathcal{R}_m$ and $\mathcal{W}_m$ as given in Equation 1, where h, w are the dimensions of the feature map.

$$
\begin{aligned}
[\hat{C}]_{m,n} = & \frac{1}{|\mathcal{W}_m|} \sum_{i \in \mathcal{W}_m} \sum_{w,h} a_i^n(w,h) \\
& - \frac{1}{|\mathcal{R}_m|} \sum_{i \in \mathcal{R}_m} \sum_{w,h} a_i^n(w,h)
\end{aligned}
\tag{1}
$$

Since, in a convolutional layer, an activation map is computed from a single unit, we aggregate the activation map a_i^n across the spatial dimensions, hence the summations over w, h in (1). The final $[C]_{m,n}$ is $[\hat{C}]_{m,n}$ normalized by the sum of all activation values:

$$
[C]_{m,n} = [\hat{C}]_{m,n} / \sum_{m,n} [\hat{C}]_{m,n} \tag{2}
$$

For each class, low scores identify units less culpable in generating errors for that specific class, i.e., units that generate discriminant features. Units with high scores on the other hand are more culpable, i.e., units that generate confusion features. We assume such culprit units to be class-specific, i.e., each class may have its own culpability vector identifying the units most/least incriminated in that class's erroneous/correct classification because a unit n may have a distinct contribution to different classes (Figure 2 step (2)).

3.2. BoosterNet Training

BoosterNet cascades with a traditional ConvNet architecture f and model parameters θ_b, and is trained to improve the performance on OOD data processed independently by network $\mathcal{A}$. BoosterNet takes as input a tensor of class-wise features Z of size $\mathcal{W} \times \mathcal{H} \times \mathcal{C}$, where $\mathcal{W}$ and $\mathcal{H}$ are the re-sized width and height of feature map, respectively, and $\mathcal{C}$ is the number of feature channels, and predicts a class output y as $y = f\ (Z \mid \theta_b)$. $\mathcal{C} = \xi + \vartheta$, where ξ and ϑ are the desired numbers of discriminant and confusion features, respectively, which can also be specified as a fraction (or percentage) of the total number of feature maps in network $\mathcal{A}$. The class-specific features Z are extracted from certain units in network $\mathcal{A}$ based on the culpability score for each unit n as given in Equation 2. Specifically, we choose $\xi\%$ of the units with the lowest culpability scores and $\vartheta\%$ of the units with the highest culpability scores to identify their corresponding discriminant and confusion features, respectively (Figure 2 (steps 3–6)). Using the ground truth information available with the training set, BoosterNet is trained to learn from those isolated predictive data features.

BoosterNet uses episodic training [42] for its optimization based on performance on virtual test domains. Specifically, in each learning iteration, we split the training features Z into meta-train Z_{tr} and meta-test Z_{te}, where Z_{tr} and Z_{te} are episodically sampled from the class-wise confusion and discriminant features, respectively (Figure 2 (step 7)). Formally, the training paradigm consists of three parts in each iteration to update θ_b:

1. The classification loss $\mathcal{L}_{\text{task}}$ is computed on meta-train features Z_{tr}, and the model parameters of BoosterNet, θ_b, are updated by a few steps of gradient descent with a learning rate of η;

$$
\hat{\theta_b} \leftarrow \theta_b \ - \ \eta \nabla_{\theta_b} \mathcal{L}_{\text{task}}\ (\theta_b\ ; Z_{tr}). \tag{3}
$$

2. The classification loss $\mathcal{L}_{\text{task}}$ is evaluated on meta-test features Z_{te}; i.e., $\mathcal{L}_{\text{task}}(\hat{\theta_b}; Z_{te})$.

3. BoosterNet parameters θ_b are updated by the gradients calculated from a combined loss of meta-train and meta-test;

$$
\theta_b \leftarrow \theta_b \ - \ \eta \nabla_{\theta_b} \left[\mathcal{L}_{\text{task}}\ (\theta_b\ ; Z_{tr}) + \mathcal{L}_{\text{task}} \left(\hat{\theta_b}\ ; Z_{te} \right) \right] \tag{4}
$$

3.3. BoosterNet Inference

At inference time, the discriminant and confusion features of a test image are extracted from network $\mathcal{A}$ and fed to BoosterNet. Since each class may have a different culpability vector, we design a smart gate using a Siamese network [3] that only activate class-specific units most relevant to the test image. We use a Siamese network to encode each class in the training data into a unique output vector (by averaging all image vectors within that class) and compute the Euclidean distance between each of these vectors and the test image's vector. The class associated with the smallest distance is then assigned as the most probable class of the test image, and thus the corresponding units of that class are activated to extract the test image discriminant and confusion features, as demonstrated in Figure 2 (steps (8–11)). It is important to mention that the Siamese network is only used as a triggering gate to provide an initial class assignment of the test image in order to facilitate the classification process of BoosterNet. We show in Section 5, ablation study 3, that BoosterNet is capable of achieving a good performance even without the Siamese network gate, but gating is shown to improve performance.

4. Quantifying Domain Shift

It is important to analyze when domain shift is likely to impact the performance of a model significantly. In order to better understand and evaluate this, we calculate the *representation shift*, R, proposed in [48], to quantify the statistical difference between the datasets in the evaluation of BoosterNet in Section 5. This metric, R, measures the differences in the distribution of layer activations of a model between datasets from two domains, capturing the model-perceived similarity between the two datasets.

We denote by $p^{\mathcal{T}}_{\mathbf{c}_{ln}}$ the continuous distribution of c_{ln} computed from the training input data $X^{\mathcal{T}} = \left\{x^{\mathcal{T}}_1, \ldots, x^{\mathcal{T}}_z\right\}$, where c_{ln} is the mean value of the activation map of each convolutional filter n_i in a layer l, and z is the number of images in $X^{\mathcal{T}}$. The test dataset, $X^{\mathcal{S}} = \left\{x^{\mathcal{S}}_1, \ldots, x^{\mathcal{S}}_q\right\}$ where q is the number of images in $X^{\mathcal{S}}$, similarly generates $p^{\mathcal{S}}_{\mathbf{c}_{ln}}$. The representation shift R is then defined as the mean discrepancy D between the distributions over all filters n in a layer l;

$$R\left(p^{\mathcal{T}}, p^{\mathcal{S}}\right) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{D}\left(p^{\mathcal{T}}_{\mathbf{c}_{ln}}, p^{\mathcal{S}}_{\mathbf{c}_{ln}}\right) \tag{5}$$

where D is an arbitrary discrepancy/distance metric between $p^{\mathcal{T}}_{\mathbf{c}_{ln}}$ and $p^{\mathcal{S}}_{\mathbf{c}_{ln}}$ that tends towards zero when the two datasets' distributions are similar. That is, if the training and test datasets ($X^{\mathcal{T}}$ and $X^{\mathcal{S}}$) are statistically similar or are mapped to similar representations by the core model, the feature responses should be similar and $R\left(p^{\mathcal{T}}, p^{\mathcal{S}}\right)$ small. The basic idea is that the distributions across the two datasets are likely to be similar (i.e., small distances) if the model had indeed learnt domain-invariant features. If this is not the case, then the representation of the first dataset depends on features not present in the second dataset, likely caused by domain shift. We demonstrate in Section 5 that even a small statistical shift between training and test datasets significantly degrades the performance of the network.

5. Experiments and Results

We demonstrate the DG performance of BoosterNet on two very different benchmark datasets. We also carry out detailed ablation studies to quantify the impact of the different components of BoosterNet.

5.1. Datasets

We evaluate performance within two applications. Our first evaluation application is digit recognition, where we use the Digits-DG benchmark data consisting of four datasets: MNIST [23], MNIST-M [11], SYN [11], and USPS [5], each of which represents 10 classes from different domains. The four datasets mainly differ in font style, background and image quality. We use the original train-validation-split in each dataset throughout all our experiments. We use the training set of MNIST for training network $\mathcal{A}$, and evaluate models on all other test domains.

Our second evaluation application is skin lesion classification where we use medical benchmark data consisting of five publicly available skin lesion image datasets: HAM10000 (HAM) [50], Dermofit (DMF) [1], Derm7pt (D7P) [22], MSK [15] and UDA [15], each comprising of real patient skin lesion images collected at different clinical sites using different equipment. Each of the six datasets contained a subset of seven classes. We partition each dataset into 50% training, 20% validation, and 30% test sets and we discard data for classes beyond their 7 common classes. We use the training set of HAM for training network $\mathcal{A}$, and evaluate models on all other test set domains.

5.2. Implementation Details

For network $\mathcal{A}$, we experimented with multiple core networks that are literature standards including ResNet-18 (RN18), ResNet-50 (RN50), ResNet-152 (RN152), and VGG-16. We trained each model using cross-entropy loss for 100 epochs, a constant learning rate of 1e-5 and a batch size of 32. As for BoosterNet, the architecture was a regular ConvNet with conv-pool-conv-pool-fc-fc-softmax layers. We opted this architecture for simplicity and fairer comparison with SOTA. BoosterNet was trained using cross-entropy loss for 50 epochs, a constant learning rate of 1e-5 and a batch size of 8. Finally, the Siamese gate had a ConvNet architecture of conv-pool-conv-pool-conv-pool-fc-softmax in each branch. It was trained using a supervised contrastive loss for 100 epochs, a constant learning rate of 1e-5 and a batch size of 32. For the skin lesions benchmark, we balanced the classes within each dataset since they suffered from extreme class-imbalance. To simulate realistic training approaches, we augmented the data by resizing all images to 650 × 650, randomly resizing and cropping 224 × 224, and randomly flipping and rotating $[90°, 180°, 270°]$. The validation and test images were resized to 650 × 650.

5.3. Evaluation on Digits

Effect of domain shift on network $\mathcal{A}$ performance. In Exp$\mathcal{A}$-$\mathcal{D}$, Table 1, we evaluate the performance of network $\mathcal{A}$ on in-distribution target data (i.e., MNIST test set) as well as OOD target data (i.e., other domain test sets). We observed that all four core models performed well when tested on in-distribution test data, achieving a classification accuracy of 99.34% in Exp$\mathcal{B}$ (best case scenario). However, the performance on other domain test data was much worse, dropping below 10% in some cases, due to domain shift. To quantify the shift between different domain test sets, we measured the representation shift score R calculated from the last convolutional layer using the Wasserstein distance as a discrepancy metric. We reported the values in brackets in Table 1. The R value between training and test sets of the same domain (e.g., MNIST/MNIST) is smaller than that of different domains. We observed, in general, a clear negative correlation between R and classification accuracy, as expected. This confirmed that in contrast to a human classifier who would not be tripped by such small statistical domain shifts, all SOTA network architectures tested have

Table 1. Evaluation on Digits: Test set classification results of experiments $\mathcal{A}$ - $\mathcal{S}$. (–) indicates that BoosterNet is not used. ξ and ϑ are set to 15% and 35%, respectively. ET and CT are abbreviations of episodic and conventional training, respectively. The training set in all experiments is MNIST train set.

	Experimental Setup		**Classification Accuracy % in Test Datasets (R Shift Values)**				**Avg Classification ± std (%)**
Exp	Network $\mathcal{A}$	BoosterNet Training	MNIST	MNIST-M	SYN	USPS	
Baselines							
$\mathcal{A}$	RN18	–	98.81 (0.0201)	16.95 (0.425)	8.13 (0.911)	18.42 (0.295)	35.57±42.39
$\mathcal{B}$	RN50	–	99.34 (0.0165)	18.3 (0.415)	11.24 (0.836)	19.73 (0.351)	37.15±41.62
$\mathcal{C}$	RN152	–	98.01 (0.0216)	15.85 (0.533)	7.62 (0.981)	17.45 (0.472)	34.73±2.40
$\mathcal{D}$	VGG16	–	85.95 (0.0381)	13.22 (0.797)	6.89 (1.186)	13.88 (0.502)	29.98±37.44
BoosterNet (proposed)							
$\mathcal{E}$	RN18	ET	99.01 (0.0182)	76.21 (0.189)	51.06 (0.304)	83.62 (0.088)	77.47±20.01
$\mathcal{F}$	RN50	ET	**99.72** (0.017)	**77.89** (0.097)	54.39 (0.299)	**84.31** (0.081)	**79.07±8.83**
$\mathcal{G}$	RN152	ET	98.64 (0.022)	74.22 (0.214)	49.71 (0.318)	82.78 (0.091)	76.33±20.43
$\mathcal{H}$	VGG16	ET	97.33 (0.030)	69.53 (0.295)	47.26 (0.346)	79.63 (0.199)	73.43±20.89
BoosterNet without Episodic Training							
$\mathcal{I}$	RN50	CT	98.15	65.98	49.31	77.53	72.74±20.52
BoosterNet without Culpability Sorting							
$\mathcal{J}$	RN50	ET	94.29	49.37	37.81	73.05	63.63±25.15
BoosterNet without Class-specific Culpability-based Feature Selection							
$\mathcal{K}$	RN50	ET	97.65	75.11	48.63	81.84	75.80±20.43
Training an End-to-End Modified Network $\mathcal{A}$							
$\mathcal{L}$	mod-RN50	CT	95.3	61.28	35.74	74.68	66.75±24.96
Comparison against SOTA							
$\mathcal{M}$	Mixup [61]	–	97.35	54.0	41.2	76.6	67.28±24.81
$\mathcal{N}$	M-ADA [42]	–	99.29	67.49	48.95	78.53	73.56±21.04
$\mathcal{O}$	JiGen [4]	–	99.14	57.48	43.26	77.35	69.31±24.31
$\mathcal{P}$	UgMG [41]	–	98.92	67.37	**57.06**	77.25	75.15±17.86
$\mathcal{Q}$	PAR [53]	–	99.31	58.14	44.67	76.17	69.57±23.65
$\mathcal{R}$	Self-super [17]	–	98.98	58.15	41.92	77.1	69.03±24.59
$\mathcal{S}$	CCSA [37]	–	98.94	49.29	37.31	83.72	67.31±28.83

failed to handle the domain shift and could not generalize.

Performance of BoosterNet. In Exp$\mathcal{E}$-$\mathcal{H}$, we appended BoosterNet to each core network from Exp$\mathcal{A}$-$\mathcal{D}$. Through episodic training, we trained BoosterNet with the corresponding class-specific discriminant and confusion features extracted from ξ=15% of the units with the lowest culpability scores and ϑ=35% of the units with the highest culpability scores. ξ and ϑ were empirically chosen. The classification results of BoosterNet on test sets are given in Table 1. We observed significant improvement in performance across all datasets using BoosterNet, demonstrating the effectiveness of our framework in improving generalization capabilities by learning from network $\mathcal{A}$'s culpable features. In addition, we noticed that, with BoosterNet, R values across domains are smaller which helped maintaining a higher accuracy. For all remaining experiments, we used RN50 as the core architecture of network $\mathcal{A}$ as it had the best average performance.

Ablation Studies. The three main components in BoosterNet are: 1) episodic training, 2) culpability-based sorting, and 3) class-specific feature selection. In Exp$\mathcal{I}$-$\mathcal{K}$, we studied the effect of each component on BoosterNet performance:

1. **Validation of episodic training scheme:** In Exp$\mathcal{I}$ Table 1, we evaluated the performance of BoosterNet without incorporating the episodic training scheme, i.e., through conventional training. By comparing with Exp$\mathcal{F}$, we observed as expected that without episodic training, accuracy dropped by ∼6.3% on average demonstrating that episodic training effectively improves the generalizability of BoosterNet to OOD data. Yet, even with a conventional training, BoosterNet had better generalization performance compared to baseline in Exp$\mathcal{B}$.

2. **Validation of sorting culpability-based features:** In Exp$\mathcal{J}$, we validated the efficacy of the discriminant and confusion features by training BoosterNet on a set of random features, i.e., without incorporating the culpability measure. Specifically, we selected random units in network $\mathcal{A}$ and extracted the corresponding random features, keeping the same size of the training data features as in Exp$\mathcal{F}$. For the episodic training, meta-train and meta-test sets were randomly sampled from the training data features. We present the classification results on test sets in Table 1. We observed a sharp drop in performance compared to results in Exp$\mathcal{F}$, ∼15%, confirming that not all features contribute positively to the generalization capability of a model or are relevant to improving the final predictions.

3. **Validation of class-specific culpability-based feature selection:** In Exp$\mathcal{K}$, we gauged the effect of

class-specific features on training BoosterNet. Specifically, we identified units in network $\mathcal{A}$ contributing to the *overall* correct or incorrect predictions across all classes, then we extracted the corresponding features and fed them to BoosterNet. Indeed, we discovered that 18% of the units across all ξ sets are the same whereas 23% of the units are common across all ϑ sets. At inference, we discarded our Siamese gate and we deployed these common units to extract test image features. The classification results on the test sets are shown in Table 1. While the performance was better compared to Exp$\mathcal{B}$ (baseline), discarding the class-specificity feature selection resulted in performance drop compared to Exp$\mathcal{F}$ by ~3%, highlighting the usefulness of our class-specific feature selection. Though discarding class-specific selection is simpler as no gating mechanism is needed and the units are always fixed, we believe that it decreases reliability as performance will be strongly dependent on correlations between the different classes in the training dataset.

Comparison against an end-to-end modified baseline. In Exp$\mathcal{L}$, instead of cascading BoosterNet with a core network, we modified the core network architecture (RN50) by removing the last layer and appending it with BoosterNet (c.f. mod-RN50). We trained mod-RN50 in an end-to-end manner on MNIST using cross-entropy loss for 100 epochs, a constant learning rate of 1e-5 and a batch size of 32. We observed an improved performance of mod-RN50 compared to RN50 baseline in Exp$\mathcal{B}$, but performance trailed behind that of using BoosterNet as a separate network (Exp$\mathcal{F}$, Table 1).

Comparison against SOTA DG methods. We compared our proposed approach with existing DG methods from different categories: 1) Data augmentation: Mixup [61] (Exp$\mathcal{M}$), M-ADA [42] (Exp$\mathcal{N}$), JiGen [4] (Exp$\mathcal{O}$), and UgMG [41] (Exp$\mathcal{P}$), 2) Adversarial training: PAR [53] (Exp$\mathcal{Q}$) and Self-super [17] (Exp$\mathcal{R}$), 3) Feature alignment: CCSA [37] (Exp$\mathcal{S}$). We report the results in Table 1. Clearly, BoosterNet outperforms all methods except that UgMG on SYN is better by ~3%, yet the average performance across all test domains is lower than that of BoosterNet. The worst performance belongs to Mixup and CCSA, since the generation of training pairs in the former is conducted only in a convex manner whereas the contrastive loss in the latter is applied to a single training domain but usually requires a number of training domains to avoid overfitting.

5.4. Evaluation on Skin Lesions

Repeating the same experiments. We repeated all experiments on our second benchmark data, skin lesions, and reported results in Table 2. We noted a similar tendency (i.e., negative correlation) between R values across domains and baseline network performance. However, BoosterNet suffered less and improved generalization significantly compared to baselines (Exp$\mathcal{A}$-$\mathcal{D}$ vs Exp$\mathcal{E}$-$\mathcal{H}$). In the remaining skin-based experiments, we used RN18 as the core architecture of network $\mathcal{A}$ as it had the best average performance. From Table 2, we observed that without incorporating any of the three components of BoosterNet, i.e., episodic training (Exp$\mathcal{I}$), culpability-based feature sorting (Exp$\mathcal{J}$), and class-specific selection (Exp$\mathcal{K}$), the performance dropped compared to Exp$\mathcal{E}$ by 6.2%, 25.2% and 9.7%, respectively. To further evaluate performance, we compared BoosterNet against three DG methods that were previously shown to be capable to generalize across OOD domains on medical imaging data: a modified CCSA [58] (Exp$\mathcal{M}$), MixUP [61] (Exp$\mathcal{N}$) and LDDG [29] (Exp$\mathcal{O}$). BoosterNet outperformed SOTA in all OOD domains.

Effect of changing ξ, ϑ. We studied the effect of varying the two empirically set parameters we used: the percentage of culprit units with the lowest and highest culpability scores (c.f. ξ% and ϑ%, respectively). Left and middle plots in Figure 3 report the classification performance of BoosterNet on in-distribution (HAM) and OOD validation sets, trained with different ξ & ϑ percentage values (for OOD test sets, we average the results across the four OOD domains). Comparing the results to baseline in Exp$\mathcal{A}$, we observed two interesting results: 1) training BoosterNet with *discriminant features only* (c.f. 1st row) improved results on HAM more than it did on averaged OOD sets, yet in both cases the improvements were minimal, and 2) training with *confusion features only* (c.f. 1st column) reduced the performance of BoosterNet on HAM but significantly improved it on OOD. From these results, we concluded that *learning from confusion features is more critical to achieving higher generalization performance on OOD domains, but discriminant features are still needed to balance the learning and avoid forgetting some predictive characteristics belonging to in-distribution data*. From the validation results, we end up setting ξ and ϑ to 20% and 30%, respectively, through all experiments on skin lesions benchmark, and these values proved to be the best when evaluating BoosterNet on all domain test sets (averaging in-distribution and OOD) as shown in the right plot in Figure 3. *Note: similar analysis was conducted when choosing ξ% and ϑ% for the digits experiments.*

Comparison against cascaded BoosterNets. In a final experiment, we attempt to answer this question: Can we boost a BoosterNet? For that, we investigated the effect of cascading a number of BoosterNets , such that each module aims to improve the performance of the previous one. We chose a unified architecture for all branches (regular ConvNet) and followed the same procedure for extracting discriminant and confusion features from the previous

Table 2. Evaluation on Skin Lesions: Test set classification results of experiments $\mathcal{A}$ - $\mathcal{O}$. (–) indicates that BoosterNet is not used. ξ and ϑ are set to 20% and 30%, respectively. ET and CT are abbreviations of episodic and conventional training, respectively. The training set in all experiments is HAM train set.

	Experimental Setup		**Classification Accuracy % in Test Datasets (R Shift Values)**					**Avg Classification ± std (%)**
Exp	Network $\mathcal{A}$	BoosterNet Training	HAM	DMF	D7P	MSK	UDA	
Baselines								
$\mathcal{A}$	RN18	–	83.75 (0.0136)	30.86 (0.9895)	39.68 (0.892)	48.91 (0.626)	59.32 (0.617)	52.50±18.26
$\mathcal{B}$	RN50	–	82.12 (0.0351)	31.45 (0.332)	37.81 (0.204)	45.71 (0.146)	58.07 (0.136)	51.03±17.90
$\mathcal{C}$	RN152	–	81.92 (0.0413)	28.46 (1.32)	38.36 (1.09)	43.9 (0.979)	57.25 (0.758)	49.98±18.48
$\mathcal{D}$	VGG16	–	79.15 (0.092)	24.94 (1.43)	33.82 (1.17)	50.21 (0.901)	63.57 (0.882)	50.33±19.61
BoosterNet (proposed)								
$\mathcal{E}$	RN18	ET	**85.54** (0.081)	**72.64** (0.197)	**62.51** (0.341)	**68.39** (0.281)	**79.58** (0.151)	**73.71±8.11**
$\mathcal{F}$	RN50	ET	84.14 (0.093)	71.94 (0.201)	62.01 (0.362)	67.81 (0.310)	76.9 (0.177)	72.56±7.57
$\mathcal{G}$	RN152	ET	81.64 (0.099)	67.38 (0.295)	61.89 (0.400)	66.26 (0.351)	77.34 (0.163)	70.91±7.37
$\mathcal{H}$	VGG16	ET	80.2 (0.146)	68.17 (0.313)	60.35 (0.454)	63.69 (0.392)	76.24 (0.196)	69.73±7.46
BoosterNet without Episodic Training								
$\mathcal{I}$	RN18	CT.	76.34	68.28	57.95	62.08	72.83	67.49±6.74
BoosterNet without Culpability Sorting								
$\mathcal{J}$	RN18	ET	69.84	35.26	38.67	33.7	65.18	48.53±15.64
BoosterNet without Class-specific Culpability-based Feature Selection								
$\mathcal{K}$	RN18	ET	80.61	53.25	51.95	63.27	71.05	64.02±10.83
Training an End-to-End Modified Network $\mathcal{A}$								
$\mathcal{L}$	mod-RN18	CT	79.63	42.57	44.8	56.32	70.51	58.76±14.39
Comparison against SOTA								
$\mathcal{M}$	Modified CCSA [58]	–	76.94	31.42	52.83	45.59	49.67	51.29±14.76
$\mathcal{N}$	MixUp [61]	–	78.13	34.41	40.92	31.72	43.13	45.66±16.75
$\mathcal{O}$	LDDG [29]	–	75.24	29.34	52.47	48.87	57.01	52.58±14.74

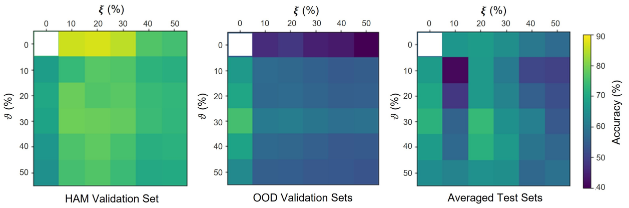

Figure 3. Performance of BoosterNet trained on different pairs of ξ% & ϑ% on in-distribution validation set (left), averaged OOD validation set (middle), and averaged test sets (right).

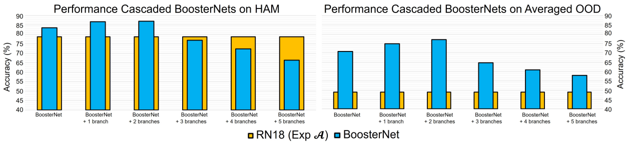

Figure 4. Performance of cascaded BoosterNets on in-distribution and OOD validation sets. RN18 in Exp$\mathcal{A}$ is used as a reference.

network, with ξ=20% and ϑ=30% fixed. The classification accuracy achieved on HAM and average OOD test sets is shown in Figure 4. Though we observed an improved performance on in-distribution and OOD sets with cascaded BoosterNets, improvements were not as significant as after the first BoosterNet, eventually resulting in 'diminishing returns' with three, possibly due to training becoming harder to achieve.

Performance of Siamese gate alone. We further quantified the performance of the Siamese network gate in comparison to the correct ground truth class and the class predicted by BoosterNet. The classification accuracy of the gate was 65.8% & 56.7%, which is ∼13% & 17% *worse* than BoosterNet results in Exp$\mathcal{F}$ on MNIST & Exp$\mathcal{E}$ on skin lesions, respectively. We argue that while Siamese net may learn separable class representations, these are not generalizable enough to OOD data. Nonetheless, as a gating mechanism, Siamese network remained useful, as it captured class-specific patterns beyond the common features across classes, as observed in ablation study 3.

6. Conclusions

We proposed BoosterNet, a simple yet effective add-on network capable of improving generalization capabilities of arbitrary core DNNs via learning from discriminant and confusion features extracted from the core network using a unit culpability criterion that measures the contribution of each unit towards erroneous predictions in each class. BoosterNet does not require changes of the core network architecture or learning scheme, making it ideal for non-expert DNN users in practical real-world applications. Through a comprehensive set of experiments, we validated BoosterNet on benchmarks data from two application areas and showed improved generalization to OOD domains compared to baselines and SOTA. One limitation in our work is that training should contain all the possible classes; otherwise BoosterNet would not be able to classify a test image to a new unseen class. Future work includes improving performance where BoosterNet can abstain if a test image belongs to a class that the networks has *not* seen before and assign it to *unknown class*.

References

[1] Lucia Ballerini, Robert B Fisher, Ben Aldridge, and Jonathan Rees. A color and texture based hierarchical K-NN approach to the classification of non-melanoma skin lesions. In *Color Medical Image Analysis*, pages 63–86. Springer, 2013. 5

[2] Nourhan Bayasi, Ghassan Hamarneh, and Rafeef Garbi. Culprit-prune-net: Efficient continual sequential multi-domain learning with application to skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 165–175. Springer, 2021. 3

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a 'siamese' time delay neural network. *Advances in Neural Information Processing Systems*, pages 737–737, 1994. 4

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 6, 7

[5] John S Denker, WR Gardner, Hans Peter Graf, Donnie Henderson, Richard E Howard, W Hubbard, Lawrence D Jackel, Henry S Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. In *Advances in Neural Information Processing Systems*, pages 323–331. Citeseer, 1989. 5

[6] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32:6450–6461, 2019. 2

[7] Yingjun Du, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*, 2020. 1

[8] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021. 1

[9] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217, 2021. 1, 2

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 2

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 5

[12] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1

[13] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1414–1430, 2016. 2

[14] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020. 2

[15] David Gutman, Noel C. F. Codella, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Nabin K. Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *arXiv*, abs/1605.01397, 2016. 5

[16] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 1

[17] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, page 15637–15648. Citeseer, 2019. 6, 7

[18] Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 2020. 1

[19] Shishuai Hu, Zehui Liao, Jianpeng Zhang, and Yong Xia. Domain and content adaptive convolution for domain generalization in medical image segmentation. *arXiv preprint arXiv:2109.05676*, 2021. 1

[20] Chao Huang, Zhangjie Cao, Yunbo Wang, Jianmin Wang, and Mingsheng Long. Metasets: Meta-learning on point sets for generalizable representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8863–8872, 2021. 2

[21] Mohammad Arafat Hussain, Ghassan Hamarneh, and Rafeef Garbi. Cascaded regression neural nets for kidney localization and segmentation-free volume estimation. *IEEE Transactions on Medical Imaging*, 40(6):1555–1567, 2021. 1

[22] Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546, 2018. 5

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[24] Chenxin Li, Qi Qi, Xinghao Ding, Yue Huang, Dong Liang, and Yizhou Yu. Domain generalization on medical imaging classification using episodic training with task augmentation. *arXiv preprint arXiv:2106.06908*, 2021. 2

[25] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 8300–8311, 2021. 2

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[27] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019. 2

[28] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 1, 2

[29] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex C Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 2020. 7, 8

[30] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2021. 1, 2

[31] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 1

[32] Xiangtai Li, Hao He, Xia Li, Duo Li, Guangliang Cheng, Jianping Shi, Lubin Weng, Yunhai Tong, and Zhouchen Lin. Pointflow: Flowing semantics through points for aerial image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4217–4226, 2021. 1

[33] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision*, pages 624–639, 2018. 2

[34] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019. 1

[35] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021. 2

[36] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 1

[37] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017. 2, 6, 7

[38] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via IBN-net. In *Proceedings of the European Conference on Computer Vision*, pages 464–479, 2018. 2

[39] 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 International Conference on Computer Vision*, pages 1406–1415, 2019. 2

[40] Xingchao Peng and Kate Saenko. Synthetic to real adaptation with generative correlation alignment networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1982–1991, 2018. 2

[41] Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2021. 2, 6, 7

[42] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 2, 4, 6, 7

[43] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning*, pages 5389–5400, 2019. 1

[44] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in Neural Information Processing Systems*, 2021. 1

[45] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *European Conference on Computer Vision*, pages 68–83. Springer, 2020. 1, 2

[46] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10031, 2019. 1, 2

[47] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9624–9633, 2021. 2

[48] Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE Journal of Biomedical and Health Informatics*, 25(2):325–336, 2020. 4

[49] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8725–8735, 2020. 1

[50] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 3, 5

[51] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7980–7989, 2019. 1

[52] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018. 2

[53] Haohan Wang, Songwei Ge, Eric P Xing, and Zachary C Lipton. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, page 10506–10518. Citeseer, 2019. 6, 7

[54] Jindong Wang, Yiqiang Chen, Wenjie Feng, Han Yu, Meiyu Huang, and Qiang Yang. Transfer learning with dynamic distribution adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–25, 2020. 2

[55] Shujun Wang, Lequan Yu, Kang Li, Xin Yang, Chi-Wing Fu, and Pheng-Ann Heng. DoFE: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets. *IEEE Transactions on Medical Imaging*, 39(12):4237–4248, 2020. 1

[56] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A Fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021. 2

[57] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2021. 1, 2

[58] Chris Yoon, Ghassan Hamarneh, and Rafeef Garbi. Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 365–373, 2019. 1, 2, 7, 8

[59] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2100–2110, 2019. 2

[60] Zhenyu Yue, Fei Gao, Qingxu Xiong, Jun Wang, Teng Huang, Erfu Yang, and Huiyu Zhou. A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition. *Cognitive Computation*, 13(4):795–806, 2021. 1

[61] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2, 6, 7, 8

[62] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5372–5382, 2021. 2

[63] Yifan Zhang, Ying Wei, Qingyao Wu, Peilin Zhao, Shuaicheng Niu, Junzhou Huang, and Mingkui Tan. Collaborative unsupervised domain adaptation for medical image diagnosis. *IEEE Transactions on Image Processing*, 29:7834–7844, 2020. 1

[64] 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

[65] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020. 1

[66] Danbing Zou, Qikui Zhu, and Pingkun Yan. Unsupervised domain adaptation with dual-scheme fusion network for medical image segmentation. In *The International Joint Conference on Artificial Intelligence*, pages 3291–3298, 2020. 1