# Domain Generalization via Feature Variation Decorrelation

Chang Liu
Northeastern University
Boston, MA, USA
liu.chang6@northeastern.edu

Lichen Wang
Northeastern University
Boston, MA, USA
wanglichenxj@gmail.com

Kai Li
Northeastern University
Boston, MA, USA
kaili@ece.neu.edu

Yun Fu
Northeastern University
Boston, MA, USA
yunfu@ece.neu.edu

## ABSTRACT

Domain generalization aims to learn a model that generalizes to unseen target domains from multiple source domains. Various approaches have been proposed to address this problem by adversarial learning, meta-learning, and data augmentation. However, those methods have no guarantee for target domain generalization. Motivated by an observation that the class-irrelevant information of sample in the form of semantic variation would lead to negative transfer, we propose to linearly disentangle the variation out of sample in feature space and impose a novel class decorrelation regularization on the feature variation. By doing so, the model would focus on the high-level categorical concept for model prediction while ignoring the misleading clue from other variations (including domain changes). As a result, we achieve state-of-the-art performances over all of widely used domain generalization benchmarks, namely PACS, VLCS, Office-Home, and Digits-DG with large margins. Further analysis reveals our method could learn a better domain-invariant representation, and decorrelated feature variation could successfully capture semantic meaning.

## CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; **Neural networks**; *Image representations*.

## KEYWORDS

domain generalization, deep learning, transfer learning, Out-of-distribution

**ACM Reference Format:**
Chang Liu, Lichen Wang, Kai Li, and Yun Fu. 2021. Domain Generalization via Feature Variation Decorrelation. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475311


*MM '21, October 20–24, 2021, Virtual Event, China*

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475311

## 1 INTRODUCTION

Deep learning has made significant progress in various fields in the past few years, such as computer vision [9, 19, 20, 26] and natural language processing [3, 14]. The experimental setup of deep learning assumes that data is independent and identically distributed. In other words, training and testing data are drawn from the same data distribution. However, this assumption does not always hold in reality, as the distribution of training and testing data may differ drastically. In this case, deep models often perform poorly due to the domain distribution shift. Considering the expensiveness and difficulty of collecting data from various distributions, enhancing the generalization ability of deep models is important, and therefore Domain Generalization (DG) is proposed to learn generalizable models by leveraging information from single or multiple source domains as shown in Figure 1.

Various methods for domain generalization have been introduced. Many existing DG methods aim at learning a source domain-invariant feature representation such as adversarial learning [18, 21] and Explicit feature alignment [37]. However, without access to any target domain data, the model learned with domain alignment can still overfit the source domains. Alternatively, meta-learning methods [1, 16] have received lots of attention by simulating the domain shift with held-out source domain during training. Similarly, meta-learning still ais at narrowing down the domain gap between source domains but has no guarantee for target domain generalization. Additionally, data augmentation [36, 42] is an effective direction to enrich the diversity of training distribution towards newly introduced domains.

Motivated by an observation that class-irrelevant information, such as semantic variation (e.g., geometric deformation, background changes, simple noise, domain shift), would lead to negative transfer if it is served as a clue for class prediction, we target at capturing the invariance between source domains by decorrelating the sample variations to class information. By doing so, the deep model would ideally focus on the high-level categorical concept of objects for prediction, while sample variations would not affect the model decision.

Based on the finding that semantic relationships between samples can be captured by the spatial positions of their deep features [34, 38], we propose to disentangle the semantic variation out of sample in feature space by linearly subtracting the feature vector with its corresponding online estimated class prototype. We demonstrate that our estimated class prototype captures the meaningful

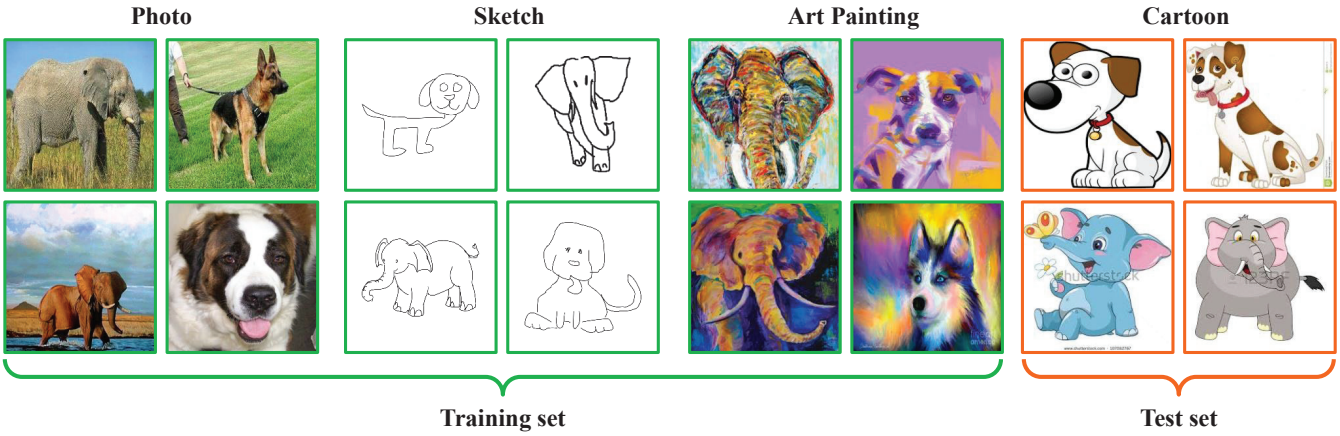


**Figure 1: Examples from the dataset PACS [15] for Domain Generalization (DG). The training set consists of the domains of sketch, photos, and art paintings. The goal of DG is to learn a model generalize well on unseen domain of cartoon.**

categorical information in Figure 4 (1) and the disentangled variation vectors with similar semantic meaning, such as "standing toward left" and "frontal view," are close to each other in feature space in Figure 4 (2). Finally, we propose **Feature variation Decorrelation** loss to decorrelate the variation vectors to categorical information.

In summary, our main contributions are shown as follows:

- To our best knowledge, we conduct the first attempt in domain generalization to linearly disentangle the semantic variation out of sample in feature space and utilize it to learn a domain-invariant representation via category decorrelation.
- We propose a novel category decorrelation regularization on feature variation in the form of conditional entropy maximization.
- We have done qualitative analysis to visualize the feature distribution of our trained model, and the nearest neighbors of our estimated class prototype, and feature variation.
- We conduct extensive quantitative ablation study and experiments on four Domain generalization benchmarks, namely PACS, VLCS, OfficeHome, Digits-DG, and verify that our method can outperform previous state-of-the-arts by a large margin.

## 2 RELATED WORK

Existing domain generation (DG) methods can mainly be categorized into the following three groups, methods based on meta learning, method based on data augmentation, and methods based on domain-invariant representation learning.

### 2.1 Meta-learning

Meta-learning-based methods simulate the domain shift scenario during training to enhance the robustness of the model against unseen domains. Li et al. [16] provides a general framework for meta-learning-based DG by back-propagating the second-order gradients calculated on a random meta-test domain split from the source domains at each iteration. Further, [6] applies a shared feature extractor with several domain-specific aggregation modules, which are aggregated during inference to predict the class label. [17] trains independent feature extractors and classifiers for each source domains and improve the robustness of model by training them in an episodic manner. However, meta-learning methods do not always have the guarantee for target domain generalization as the simulated domain they trained on might not cover the target domain shifts.

### 2.2 Data augmentation

Data Augmentation enhances the generalization performance of a model by the transformation of data [10]. Typical augmentations include Gaussian noise, random color distortions, rotations and so on. Other than typical augmentation techniques, [33] is the first work applying domain randomization to generate new data which could simulate the complex environments based on the training samples. By adjusting the texture, shape of objects, and the illumination or camera angle to simulate the environments, domain randomization usually performs better than traditional augmentation techniques in domain generalization. Additionally, adversarial data augmentation [42] is another important branch to optimize the augmented training samples in terms of minimizing the generalization error while maintaining the reliability via controlling the tolerance rate. On top of this, Mixup [39] is adopted in several DG methods to get new samples in either the original space or in the feature space by conducting linear interpolation between two randomly chosen instances and their labels. While these generated domains differ significantly from the source domains, they potentially do not reflect practical domain differences.

### 2.3 Domain-invariant representation learning

The goal of domain-invariant representation learning based methods is to reduce the discrepancy of representation with respect to different domains while retraining the discriminative class information. [11] uses a kernel-based method to minimizes mean domain discrepancy while maximizes mean class discrepancy. [18] adopts adversarial auto-encoders to align the representations from

all the source domains using adversarial learning. [31] uses domain-specific normalizations to explore the best combination of batch and instance normalization such that domain-agnostic representations can be learned. [2] learns the domain invariant representation by solving jigsaw puzzles. [12] iteratively discards the network units with the highest gradients while using the remaining units to learns useful features. [37] provides extrinsic supervision in the form of a metric learning task and intrinsic supervision in terms of a self-supervised auxiliary task.

Our method is aligned with domain-invariant representation learning and feature disentanglement. Different from [30] which proposes a low-rank decomposition on the final classification layer, we conduct a linear disentanglement on variation in feature space and propose a decorrelation regularization. Unlike the traditional feature disentanglement, method [29], we do not introduce an extra disentangler network to decompose features into several components and reconstructs the features via auto-encoder. Instead, we conduct disentanglement on the original feature space.

# 3 METHOD

In this section, we start with an explanation of our motivation in Sec. 3.2. Then, we introduce the feature variation disentanglement and discuss variance transfer idea in Sec. 3.3. Finally, we present our novel feature variation decorrelation loss in Sec. 3.4. Figure 2 shows the framework of the proposed method.

## 3.1 Problem Setting

Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote data and the label spaces. In domain generalization (DG), there are $K$ source domains $\{\mathcal{D}_i\}_{i=1}^{K}$ and L target domains $\{\mathcal{D}_i\}_{i=K+1}^{K+L}$. Specifically, the source domain dataset is denoted as $\mathcal{D}_i = \{(x_j^i, y_j^i)|_{j=1}^{N_i}\}$ where $x_j^i$ and $y_j^i$ denote the $j^{th}$ training sample and its label from the $i^{th}$ domain , and $N_i$ is the number of source images in domain $i$. Note that all domains share the same label space $\mathcal{Y}$. The goal of DG is to generalize the model training on the samples of source domains to unseen target domains. We define a feature extractor parameterized by $\theta$ as $F_\theta$ and classifier parameterized by $\phi$ as $C_\phi$ such that the network can be represented as $G_{\theta,\phi}(\cdot) = C_\phi \circ F_\theta(\cdot)$.

## 3.2 Motivation

Our method is motivated by an observation that class-irrelevant information, such as sample variation (e.g., geometric deformation, background changes, simple noise) and domain variation (image style changes), is sometimes served as a clue for class prediction, leading to negative transfer [29], especially when the target domain is highly heterogeneous. Therefore, we aim at proposing a method to capture the invariance between source domains by decorrelating those semantic/domain variations to class information such that the model would only focus on high-level semantic structure for prediction when unseen target samples are given.

As deep neural networks are good at linearizing features of input samples [34, 38], the semantic relationships between samples can be captured by the spatial positions of their deep features. We propose to linearly disentangle the observed variations out of samples in feature space by subtracting the feature vectors with an online estimated class prototype. Then, the disentangled variations are decorrelated to the class information by our new loss function. By doing so, our model would not suffer from making the wrong prediction based on undesired variation clues.

## 3.3 Variation Disentanglement

In this section, we first build up a feature memory bank, compute the class prototype based on the memory bank and finally obtain the Semantic Variation Disentanglement.

*3.3.1 Multi-variate Normal Distribution Assumption.* We assume that the data distribution follows a multi-variate normal distribution $\mathcal{N}(\mu_c, \Sigma_c)$ where $\mu_c$ and $\Sigma_c$ denote the class conditional mean vector and covariance matrix. In the following sections, we would estimate the class conditional mean as class prototype and obtain the feature variations.

*3.3.2 Online Memory Bank Update.* First, we warm start the model training for several epochs to make sure a meaningful feature space is obtained. Then, we extract all the source features $z_j^i = F_\theta(x_j^i)$ by our model and save them into a feature memory bank $M = \{(z_j^i, y_j, d_j)|_{i=1}^{N}\}$ where $y_j$ denotes the class label and $d_j$ denotes the domain label for feature $z_j^i$.

Note that our memory bank $M$ is updated on the fly with the latest features to replace the old ones. Formally, in each iteration k, we will update a batch of features in memory module M:

$$z_j^M \leftarrow z_j^i, \quad j \in \mathcal{B}^k \tag{1}$$

Besides exactly replacing the old features with the new ones, we also consider updating the features in a moving average manner. Specifically, the feature in memory module M will be updated with the moving average between the new feature and the old feature of last epoch:

$$z_j^t = \gamma z_j + (1-\gamma) z_j^{t-1}, \quad j \in \mathcal{B}^k, \tag{2}$$

where $\gamma$ is the moving average coefficient.

The updating rule would affect the quality of features stored in the memory bank, and thus directly relate to the quality of semantic variation disentanglement. We conduct ablation study on updating strategies in Table 5 and find out Equation (1) gives the better performance.

*3.3.3 Prototype Selection.* We estimate the class prototype by taking the average over the features from same class. Specifically, the class prototype $y$ can be represented as:

$$\hat{\mu}_y = \frac{1}{K}\sum_{d=1}^{K} \hat{\mu_y^d}, \qquad \hat{\mu}_y^d = \frac{1}{N_y^d}\sum_{j=1}^{N_y^d} z_j^i, \tag{3}$$

where $K$ is the number of source domains. $\hat{\mu}_y^d$ is domain-specific class prototype. $N_y^d$ is the number of samples in class $y$ and domain $d$. Typically, the representations of class prototype reflect the neural semantic of each class. For example, the class prototype of human faces is usually frontal with neural expression. This motivates us to obtain the semantic feature variation by simply subtracting the features to their corresponding class prototypes.

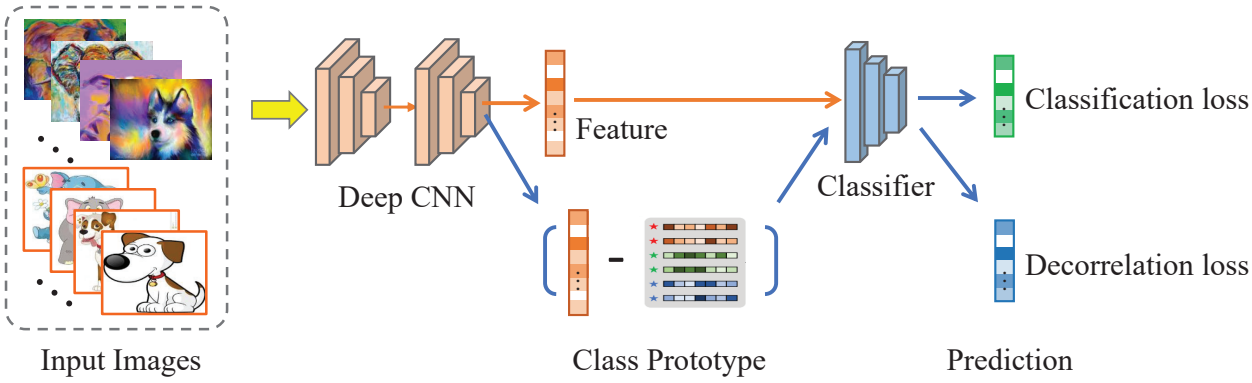


**Figure 2: Framework of our feature variation decorrelation method. In Sec. 3.3, we introduce the semantic variation defined as the displacement between feature vector and its corresponding class prototype. In Sec. 3.4, we present the feature variation decorrelation loss in the form of conditional entropy maxmization. We claim that by decorrelating the feature variations to categorical information, model could capture high-level categorical concept while ignoring other undesired clues that mislead the model prediction.**

*3.3.4 Semantic Feature Variation.* We consider the variations in the latent feature space which captures semantic information about the given sample. Formally, we denote the semantic variations $v_j$ as the displacement between an j-th feature vector $z_j$ and the estimated prototype of class $y_j$ in the feature space:

$$v_j = z_j - \hat{\mu}_{y_j}. \tag{4}$$

Those variations could represent the semantic meaning such as shape, color, visual angle and background. As we use unified class prototype for the samples of same class but different domain, the the semantic variations $v_j$ also capture domain variation. We claim those variations should not contribute to the categorical prediction and we introduce a new loss function to weaken this correlation in the next section.

**Discussion on variation transfer.** Inspired from long-tailed recognition [22] to transfer variance from head class to tail class such that the samples from tail class could be augmented, we also think about transferring variance across class and domains to augment our training samples. Instead of manipulating in original space, we augment the samples in feature space. Specifically, we adopt conditional GAN [25] to generate new features of class $y$ with variations from other class. The class prototype $\hat{\mu_y}$ and variation $v_j$ are the inputs to the generator $G$, and new augmented feature $z^j_{aug}$ with class label $y$ is generated as:

$$z^j_{aug} = G(\hat{\mu_y}, v_j). \tag{5}$$

Additionally, a discriminator $D$ is introduced to distinguish the features between real and fake (generated). The discriminator is optimized to minimize the following objective function while generator is optimized to maximize it to fool the discriminator.

$$\mathcal{L}_{adv}(G, D) = \frac{1}{N}\sum_{j=1}^{N} \log D(z^j, y_j) + \frac{1}{N}\sum_{j=1}^{N} \log D(1 - G(\hat{\mu_y}, v_j), y), \tag{6}$$

where $N$ is the number of all source samples.

We incorporate those augmented features for training and report the result of variance transfer via feature augmentation in Table 5. We found out that it achieves marginal improvement compared to vanilla model. Please check section 4.5.1 for more detail.

## 3.4 Feature Variation Decorrelation

The key idea of our method is that the observed variations of real data (e.g. geometric deformation, background changes, simple noise and domain style) should not affect the model categorical prediction such that the model would only focus on high-level categorical concept for learning and generalize better to unseen domains.

First, we train our model on samples from source domains with the following objective:

$$\min_{\mathcal{F},C} \frac{1}{N}\sum_{j=1}^{N} \mathcal{L}_{ce}\left(C(\mathcal{F}(x_j)), y_j\right), \tag{7}$$

where $\mathcal{L}_{ce}$ denotes the cross-entropy loss.

In order to decorrelate the variations from categorical information, we propose a novel Feature Variation Decorrelation (FVD) loss via pushing the classifier prediction of feature variation to be uniform distributed. Specifically, we compute the variation of a feature based on Eqation 4, feed it into the classifier, and maximize the conditional entropy of this prediction. The conditional entropy of variation prediction can be formally formulated as:

$$\mathcal{L}_{FVD} = -\frac{1}{N}\sum_{j=1}^{N} C(\mathcal{F}(x_j) - \hat{\mu_{y_j}}) \log(C(\mathcal{F}(x_j) - \hat{\mu_{y_j}})). \tag{8}$$

By maximizing $\mathcal{L}_{FVD}$, the classifier prediction of variation would be close to a uniform distributed vector. In other words, it does not have any correlation with categorical information.

Finally, the total loss function can be formulated as:

$$\mathcal{L}_{all} = \mathcal{L}_{ce} - \lambda \mathcal{L}_{FVD}, \tag{9}$$

where $\lambda$ is the hyperparameter to balance the decorrelation loss.

# 4 EXPERIMENT

In this section, we organize the experiments as: **(1)** We present quantitative evaluation results on four public domain generalization benchmarks, namely Digits-DG, PACS, VLCS and Office-Home. **(2)** We conduct extensive ablation study of our proposed idea in section 4.5. **(3)** We present the qualitative analysis in section 4.6 in terms of visualizing feature distribution, prototype and feature variation.

## 4.1 Datasets

We evaluate the proposed method on various domain generalization datasets as follows:

- Digits-DG [42] consists of four digit recognition datasets, namely MNIST [13], MNIST-M [8], SVHN [28] and SYN [8], which differ drastically in font style and background.
- PACS [15] contains 9991 images from 4 domains, i.e., Photo, Art painting, Cartoon, and Sketch, which cover huge domain gaps. Images are from 7 object classes, i.e., dog, elephant, giraffe, guitar, horse, house, and person.
- VLCS [7] includes images from five classes over four domains. The domains are defined by four image origins, i.e., images were taken from the PASCAL VOC 2007, LabelMe, Caltech and Sun datasets.
- Office-Home [35] contains 65 object categories to over 4 domains (Art, Clipart, Product, and Real-World). The domain variations mainly take place in background, viewpoint and image style.

## 4.2 Evaluation protocol

For fair comparison with prior work, we follow the leave one-domain-out protocol in [2, 4]. Specifically, one domain is chosen as the test domain while the remaining domains are used as source domains for training. The top-1 classification accuracy is used as performance measure. For each target domain, the reported results are averaged over five independent runs with different random seeds. We report the standard deviation as 0 for models which have not reported them in their respctive paper. For each dataset, we use the official source domain validation set for selecting hyperparameters if it is available. Otherwise, we split the data from source domains into training and validation sets.

## 4.3 Implementation Details

For Digits-DG, we follow the experimental setup of [42] and use their architecture for the feature extractor. Specifically, the CNN backbone is constructed with four 64-kernel 3×3 convolution layers and a softmax layer. ReLU and 2×2 max-pooling are inserted after each convolution layer. The networks are trained from scratch using SGD with initial learning rate of 0.05, batch size of 128 and weight decay of 5e-4 for 50 epochs. The learning rate is decayed by 0.1 every 20 epochs.

For PACS, We follow the experimental protocol defined in [15] and use alexnet, ResNet-18 and ResNet-50 as the CNN backbone. For VLCS, We follow the same experimental setup as mentioned in [24] and adopt alexnet as the CNN backbone. For Office-Home, We follow the experimental protocol as outlined in [6]. We utilize a ResNet-18 as backbone. For those three datasts, we utilize unified optimization with initial learning rate of 0.01 and batch size of 96 (32 images per source) for 540 epochs. The learning rate is decayed by 0.1 after 30 epochs.

We also adopted the same data augmentation as JiGen [2] , which includes random cropping, rescale, horizontal flipping, color jitter and grayscale.

| Method | Art. | Cartoon | Sketch | Photo | Avg. |
|---|---|---|---|---|---|
| AlexNet | | | | | |
| Vanilla | 66.68 | 69.41 | 60.02 | 89.98 | 71.52 |
| Jigen [2] | 67.63 | 71.71 | 65.18 | 89.00 | 73.38 |
| MMLD [24] | 69.27 | 72.83 | 66.44 | 88.98 | 74.38 |
| MetaVIB [5] | 71.94 | 73.17 | 65.94 | 91.93 | 75.74 |
| DGER [40] | 71.34 | 70.29 | 71.15 | 89.92 | 75.67 |
| EISNet [37] | 70.38 | 71.59 | 70.25 | 91.20 | 75.86 |
| Ours | **72.12±0.3** | **74.86±0.2** | **72.56±0.1** | **91.57±0.4** | **77.78** |
| ResNet-18 | | | | | |
| Vanilla | 77.65 | 73.93 | 70.59 | 95.12 | 79.32 |
| Jigen [2] | 79.42 | 75.25 | 71.35 | 96.03 | 80.51 |
| DGER [40] | 80.70 | 76.40 | 71.77 | 96.65 | 81.38 |
| MMLD [24] | 81.28 | 77.16 | 72.29 | 96.09 | 81.83 |
| EISNet [37] | 81.89 | 76.44 | 74.33 | 95.93 | 82.15 |
| L2A-OT [41] | 83.30 | 78.20 | 73.60 | **96.20** | 82.80 |
| Ours | **84.13±0.1** | **81.61±0.3** | **80.79±0.2** | 95.21±0.2 | **85.44** |
| ResNet-50 | | | | | |
| Vanilla | 80.22 | 78.52 | 76.10 | 95.09 | 82.48 |
| MASF [4] | 82.89 | 80.49 | 72.29 | 95.01 | 82.67 |
| DGER [40] | 87.51 | 79.31 | 76.30 | **98.25** | 85.34 |
| EISNet [37] | 86.64 | 81.53 | 78.07 | 97.11 | 85.84 |
| DSON [31] | 87.04 | 80.62 | 82.90 | 95.99 | 86.64 |
| Ours | **87.89±0.2** | **83.53±0.4** | **84.35±0.3** | 96.77±0.1 | **88.13** |

**Table 1: Results on PACS [15] dataset with AlexNet, ResNet-18 and ResNet-50 as backbones.**

| Method | Caltech | LabelMe | Pascal | Sun | Avg. |
|---|---|---|---|---|---|
| Vanilla [2] | 96.25 | 59.72 | 70.58 | 64.51 | 72.76 |
| Jigen [2] | 96.93 | 60.90 | 70.62 | 64.30 | 73.19 |
| MMLD [24] | 96.66 | 58.77 | 71.96 | 68.13 | 73.88 |
| MASF [4] | 94.78 | 64.90 | 69.14 | 67.64 | 74.11 |
| MetaVIB [5] | 97.37 | 62.66 | 70.28 | 67.85 | 74.54 |
| DGER [40] | 96.92 | 58.26 | 73.24 | 69.10 | 74.38 |
| EISNet [37] | 97.33 | 63.49 | 69.83 | 68.02 | 74.67 |
| Ours | **97.86±0.3** | **64.33±0.2** | **74.35±0.4** | **69.37±0.3** | **76.48** |

**Table 2: Results using AlexNet backbone on VLCS [7] dataset.**

## 4.4 Comparative Results

We compare with the following state-of-the-art methods, Jigen [2], CCSA [27], MMD-AAE [18], CrossGrad [32], L2A-OT [41], MetaVIB [5], DGER [40], EISNet [37], DSON [31] and RSC [12]. We denote "Vanilla" as our baseline which uses cross-entropy to train a shared feature extractor and classifier for all source domains in Table 1 - 4.

| Method | Artistic | Clipart | Product | Real-World | Avg. |
|---|---|---|---|---|---|
| Vanilla[41] | 58.9 | 49.40 | 74.30 | 76.24 | 64.75 |
| D-SAM [6] | 58.03 | 44.37 | 69.22 | 71.45 | 60.77 |
| Jigen [2] | 53.04 | 47.51 | 71.47 | 72.79 | 61.20 |
| MMD-AAE [18] | 56.50 | 47.30 | 72.10 | 74.80 | 62.70 |
| DSON [31] | 59.37 | 45.70 | 71.84 | 74.68 | 62.90 |
| RSC [12] | 58.42 | 47.90 | 71.63 | 74.54 | 63.12 |
| L2A-OT [41] | 60.60 | 50.10 | 74.80 | 77.00 | 65.60 |
| Ours | **62.24±0.3** | **54.38±0.2** | **76.12±0.1** | **78.64±0.2** | **67.85** |

**Table 3: Results using ResNet-18 backbone on Office-Home [35].**

| Method | MNIST | MNIST-M | SVHN | SYN | Avg. |
|---|---|---|---|---|---|
| Vanilla [41] | 95.8 | 58.8 | 61.7 | 78.6 | 73.7 |
| Jigen [2] | 96.50 | 61.40 | 63.70 | 74.00 | 73.90 |
| CCSA [27] | 95.20 | 58.20 | 65.50 | 79.10 | 74.50 |
| MMD-AAE [18] | 96.50 | 58.40 | 65.00 | 78.40 | 74.60 |
| CrossGrad [32] | 96.70 | 61.10 | 65.30 | 80.20 | 75.80 |
| L2A-OT [41] | 96.70 | 63.90 | 68.60 | 83.20 | 78.10 |
| Ours | **97.68±0.3** | **66.24±0.5** | **70.97±0.5** | **85.18±0.3** | **80.02** |

**Table 4: Results on Digits-DG [42] dataset.**

*4.4.1 Evaluation on PACS.* We compare our method against the state-of-the-art on PACS dataset. The performance with AlexNet, ResNet-18 and ResNet-50 as backbones is shown in Table 1. We summarise our findings as follows. (1) Our method outperforms the state-of-the-art methods on all three backbones by a significant margin, which demonstrates the effectiveness and robustness of our method to different network architectures. (2) Compared to other Domain-invariant representation learning methods such as Jigen [2], EISNet [37], DGER [40], and DSON [31], our method outperforms the best competitor by 1.92% on AlexNet, 2.56% on ResNet-18 and 2.29% on ResNet-50. It proves that by de-correlating the feature variation to categorical information, our method could learn a better domain-invariant representation than others. (3) Compared to the data augmentation method such as L2A-OT [41] which enriches the domain diversity of training data, our method yields large improvements by 2.44% on ResNet-18.

*4.4.2 Evaluation on VLCS.* The performance with AlexNet as backbones is reported in Table 2. The findings from section 4.4.1 still hold on VLCS. We achieve state-of-the-art results on VLCS where the domain shift is less severe than the PACS dataset. Thus, we demonstrate that our method generalizes well even when the source domains are not diverse.

*4.4.3 Evaluation on Office-Home.* The results are reported in Table 3 using a ResNet-18 backbone. Again, our method beats the second-best competitor by 2.25%. It is worth noting that our method achieves a large improvement with 4.28% on Clipart which has the largest domain gap between other source domains. This demonstrates the domain generalization of our method on a dataset with office and home-related objects.

*4.4.4 Evaluation on Digits-DG.* Digits-DG consists of four digit recognition datasets, each of which is considered as one domain. Note that the image size of Digits-DG is 32 × 32 compared to 224 × 224 from previous benchmarks. Also, we show the performance with the same backbones as competitors [41, 42] in table 4. Compared to data augmentation methods such as CrossGrad [32] and L2A-OT[41], our method achieves better performance by a large margin (+1.92%). Additionally, compared to other domain-invariant representation learning methods such as MMD-AAE and CCSA, our method makes a significant improvement on all domains by 4.92% on average. It demonstrates the effectiveness of our method on digit recognition with small image resolution via pushing the better domain-invariant representation.

## 4.5 Ablation Study

*4.5.1 Feature Variation Decorrelation vs Feature Augmentation.* As discussed in Section 3.3.4, semantic feature variation could be utilized to augment the features of one class along the direction of unseen variations from other class. This is based on the motivation that humans are capable of transferring variations from one visual class to another or from one domain to another. For example, when we see an animal that we have never seen before, we can imagine how it will look with different background and surroundings. As Table 5 reports, feature augmentation achieves a marginal improvement compared to the vanilla baseline by 1.51%. In comparison, feature variation decorrelation largely outperform feature augmentation by 4.61%. The possible reason of limited improvement from feature augmentation is that the generated features are still within the similar mode of original data distribution. In comparison, variation decorrelation can be considered as an implicit way to achieve inter-class variance transfer by regularizing the feature variations to be class-agnostic. Therefore, the sample variations could be implicitly shared across classes.

*4.5.2 Feature Variation Decorrelation vs Feature Disentanglement.* Our method also shares some similarity with representation disentanglement such as DADA [29] where a feature is disentangled into domain-invariant, domain-specific and class-irrelevant features. Different from our method, DADA disentangles the feature with additional disentangler network and uses auto-encoder to reconstruct the original feature. During the training and testing, only domain-invariant feature is used for class prediction in DADA while our method uses original feature for model prediction. In comparison, our method does not specially disentangle the feature into several components but regularizing the variation portion of original feature to be class-agnostic. We implemented DADA in Table 5 and term it as DisETG. It is reported that our feature correlation achieves a large margin over DADA by 4.75%, which demonstrates the effectiveness of our method over representation disentanglement method.

*4.5.3 Importance of Memory Bank Updating Rule.* As Section 3.3.2 presents, we introduce two online updating rule for memory bank: One is to replace the old features with current new features, the other is to replace the old ones with the moving average of current features. We term the former option as "New" and the latter as "Moving Avg" in Table 5. It is reported that using "New" as memory bank updating rule achieves the better performance than using "Moving Avg" by 1.27% for decorrelation experiment and by 0.69 %

| Moving avg | New | FeatAug | Decorrelation | DisETG [29] | Accuracy |
|---|---|---|---|---|---|
| - | - | - | - | - | 79.32 |
| - | - | - | - | ✓ | 80.69 |
| ✓ | - | ✓ | - | - | 80.43 |
| - | ✓ | ✓ | - | - | 81.12 |
| ✓ | - | - | ✓ | - | 84.17 |
| - | ✓ | - | ✓ | - | **85.44** |

**Table 5: Ablation study of our method with ResNet-18 on the PACS dataset. Note that "New" refers to the memory bank updating rule with current new features and "Moving Avg" refers to the memory bank updating rule with moving average of features detailed in Sec. 4.5.3. "FeatAug" refers to feature augmentation and "Decorrelation" refers to feature decorrelation in Sec. 4.5.1. "DisETG" refers to feature disentanglement in Sec. 4.5.2.**

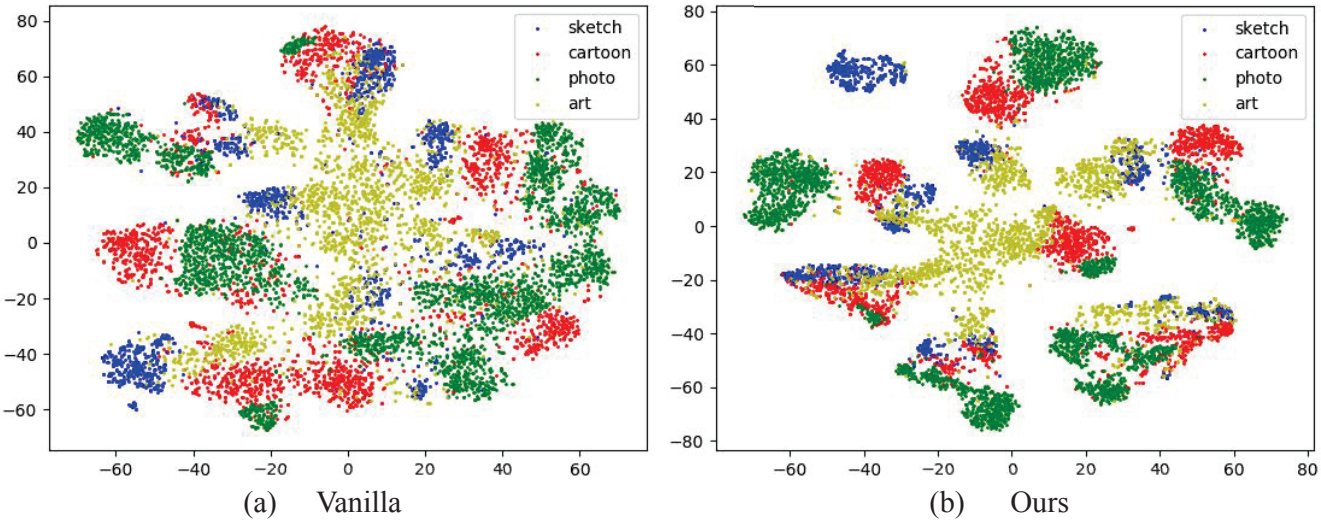


**Figure 3: The feature visualization via t-SNE on PACS with art painting as target domain. (a) Vanilla Baseline. (b) Our Feature Variation Decorrelation.**

for feature augmentation experiment. The reason might be that the moving average of feature accumulate the out-of-date features and bias the estimation of class prototype.

*4.5.4 Hyper-parameter Sensitivity .* We conduct hyperparaemter sensitivity experiment on $\lambda$ from Equation 9 and report the result in figure 5. We choose $\lambda$ from {0, 0.1, 0.5, 1, 2} for sensitivity experiment. The findings can be summarized as follows: (1) When $\lambda$ is larger than zero (which means our feature variation decorrelation loss is applied), our model has been improved drastically compared to the vanilla baseline. (2) The model achieves the best performance with 85.44% when $\lambda$ is equal to 0.1. (3) Our model is robust to the changes of hyper-parameter $\lambda$ between {0.1, 0.5, 1, 2}.

## 4.6 Qualitative Analysis

*4.6.1 Feature Visualization.* To better understand the distribution of the learned features, we exploit t-SNE [23] to analyze the feature space learned by vanilla baseline and our feature decorrelation on PACS dataset with art painting as target domain in figure 3 (a) and (b). We can qualitatively observe that our method could learn more discriminative feature space where clusters are more compact and domain-invariant than vanilla baseline.

*4.6.2 Visualization of Prototype.* To validate our prototype computed based on recorded memory feature bank could capture the high-level categorical information, we use the domain-specific prototype features to search for their nearest neighbors in feature space and visualize the neighbor samples in Figure 4(1). We can see that their nearest image neighbors successfully capture the class information and the variations of those samples are inclined to be neural and less rare.

*4.6.3 Visualization of feature variation.* To qualitatively visualize feature variations, we first compute the feature variations $v_j$ of all features by Equation 4 and randomly select some samples as anchors. We search for the nearest neighbors of anchor in feature space via feature variation vectors $v_j$ and visualize image neighbors in Figure 4(2) where each row represents the variation of corresponding anchor. The observations could be summarized as follows: (1) For each row, **the feature variation is class-agnostic** after training with our decorrelation loss. For example, the first row

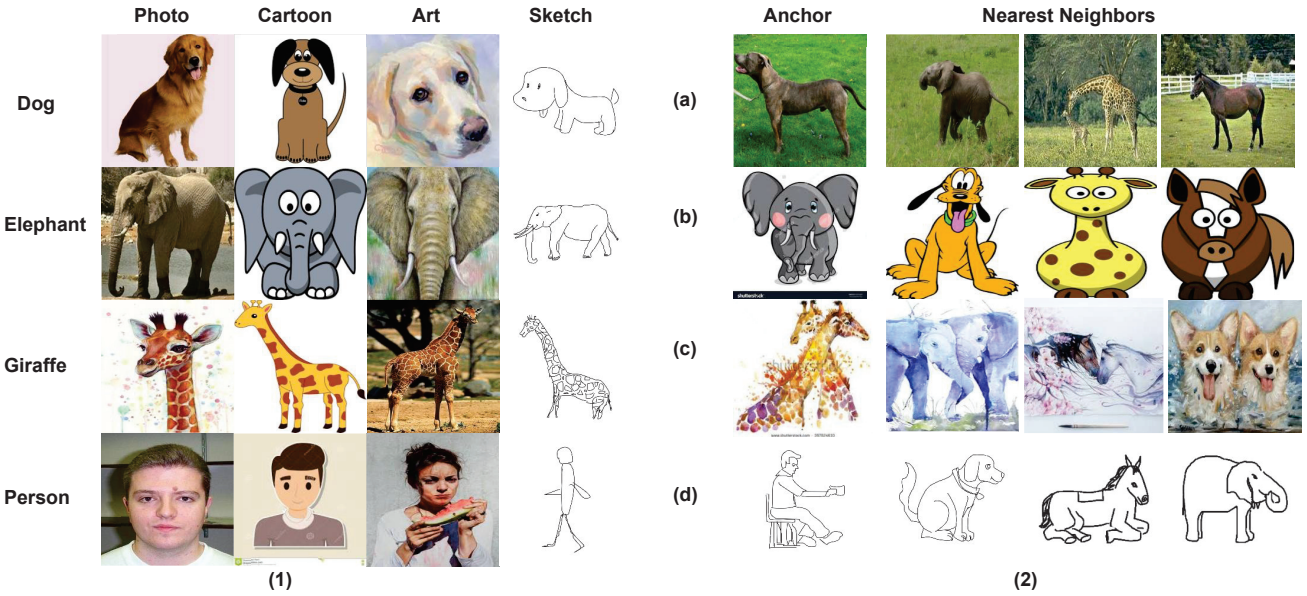


**Figure 4: (1) We visualize the prototypes' nearest image neighbors for each domain in the feature space on PACS dataset. We found out that our domain-specific prototypes computed from memory bank could capture the high-level category information. (2) To concretely visualize the feature variation, we select some samples as anchor and find its top 3 nearest image neighbors in feature space by feature variation vector. We can see that the feature variation of each row is class-agnostic and variations with similar semantic meaning are close to each other in feature space. (e.g (a) Pose towards left. (b) Frontal pose. (c) Two objects. (d) sitting towards right.)**

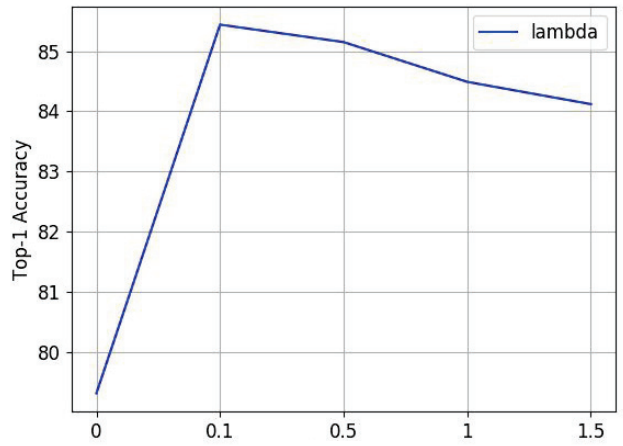


**Figure 5: Hyper-parameter sensitivity of $\lambda$ on PACS dataset.**

in Fig. 4(2.a) refers to a dog standing towards left and the image neighbors this anchor are from other classes such as elephant and horse with similar pose. (2) **feature variations with similar semantic meaning are close to each others in feature space.** For example, Fig. 4(2.a) refers to the variation of standing towards left; Fig. 4(2.b) refers to the variation of frontal pose; Fig. 4(2.c) refers to the variation of two objects; Fig. 4(2.d) refers to the variation of sitting towards right.

## 5 CONCLUSION

In this paper, we propose to linearly disentangle the variation out of sample in feature space and impose a novel class decorrelation regularization on the feature variation. By doing so, the model would focus on the high-level categorical concept for model prediction while ignoring the misleading clue from other variations (including domain changes). As a result, we achieve state-of-the-art performances over all of widely used domain generalization benchmarks, namely PACS, VLCS, Office-Home, and Digits-DG with large margins. We demonstrate that our estimated class prototype captures the meaningful categorical information and disentangled variation vectors with similar semantic meaning are close to each other in feature space.

## REFERENCES

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*. 998–1008.
[2] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2229–2238.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[4] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. 2019. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*. 6450–6461.
[5] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. 2020. Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision*. Springer, 200–216.

[6] Antonio D'Innocente and Barbara Caputo. 2018. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*. Springer, 187–198.

[7] Chen Fang, Ye Xu, and Daniel N. Rockmore. 2013. Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[8] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. 1180–1189.

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

[10] Alex Hernández-García and Peter König. 2018. Further advantages of data augmentation on convolutional neural networks. In *International Conference on Artificial Neural Networks*. Springer, 95–103.

[11] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. 2019. Domain generalization via multidomain discriminant analysis. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, Vol. 35. NIH Public Access.

[12] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. *arXiv preprint arXiv:2007.02454* (2020).

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[15] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*. 5542–5550.

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

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

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

[19] Kai Li, Martin Renqiang Min, and Yun Fu. 2019. Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3583–3592.

[20] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. 2020. Adversarial feature hallucination networks for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13470–13479.

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

[22] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. 2020. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2970–2979.

[23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[24] Toshihiko Matsuura and Tatsuya Harada. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains.. In *AAAI*. 11749–11756.

[25] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[26] Gaurav Mittal, Chang Liu, Nikolaos Karianakis, Victor Fragoso, Mei Chen, and Yun Fu. 2020. HyperSTAR: Task-Aware Hyperparameters for Deep Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

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

[29] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. 2019. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*. PMLR, 5102–5112.

[30] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. 2020. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. *arXiv preprint arXiv:2003.12815* (2020).

[31] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. 2019. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275* (2019).

[32] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. 2018. Generalizing Across Domains via Cross-Gradient Training. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1Dx7fbCW

[33] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 23–30.

[34] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. 2017. Deep feature interpolation for image content changes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7064–7073.

[35] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5018–5027.

[36] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*. 5334–5344.

[37] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. 2020. Learning from Extrinsic and Intrinsic Supervisions for Domain Generalization. *arXiv preprint arXiv:2007.09316* (2020).

[38] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. 2019. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems* 32 (2019), 12635–12644.

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[40] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. 2020. Domain Generalization via Entropy Regularization. *Advances in Neural Information Processing Systems* 33 (2020).

[41] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Learning to Generate Novel Domains for Domain Generalization. *arXiv preprint arXiv:2007.03304* (2020).

[42] Kaiyang Zhou, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. 2020. Deep Domain-Adversarial Image Generation for Domain Generalisation.. In *AAAI*. 13025–13032.