Interpolation Normalization for Contrast Domain Generalization

Mengzhu Wang
Shenzhen University
Shenzhen, China
jiannankiki@gmail.com

Junyang Chen*
Shenzhen University
Shenzhen, China
junyangchen@szu.edu.cn

Huan Wang
Huazhong Agricultural University,
Wuhan, China
hwang@mail.hzau.edu.cn

Huisi Wu
Shenzhen University
Shenzhenu, China
hswu@szu.edu.cn

Zhidan Liu
Shenzhen University
Shenzhen, China
liuzhidan@szu.edu.cn

Qin Zhang
Shenzhen University
Shenzhen, China
qinzhang@szu.edu.cn

ABSTRACT

Domain generalization refers to the challenge of training a model from various source domains that can generalize well to unseen target domains. Contrastive learning is a promising solution that aims to learn domain-invariant representations by utilizing rich semantic relations among sample pairs from different domains. One simple approach is to bring positive sample pairs from different domains closer, while pushing negative pairs further apart. However, in this paper, we find that directly applying contrastive-based methods is not effective in domain generalization. To overcome this limitation, we propose to leverage a novel contrastive learning approach that promotes class-discriminative and class-balanced features from source domains. Essentially, clusters of sample representations from the same category are encouraged to cluster, while those from different categories are spread out, thus enhancing the model's generalization capability. Furthermore, most existing contrastive learning methods use batch normalization, which may prevent the model from learning domain-invariant features. Inspired by recent research on universal representations for neural networks, we propose a simple emulation of this mechanism by utilizing batch normalization layers to distinguish visual classes and formulating a way to combine them for domain generalization tasks. Our experiments demonstrate a significant improvement in classification accuracy over state-of-the-art techniques on popular domain generalization benchmarks, including Digits-DG, PACS, Office-Home and DomainNet.

CCS CONCEPTS

• **Computer methodologies** → **Computer vision**; • **Image representations**;

KEYWORDS

Domain Generalization, Contrastive Learning, Batch Normalization

*Corresponding Author

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611841

ACM Reference Format:
Mengzhu Wang, Junyang Chen, Huan Wang, Huisi Wu, Zhidan Liu, and Qin Zhang. 2023. Interpolation Normalization for Contrast Domain Generalization. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3611841

1 INTRODUCTION

Deep neural networks (DNNs) [1, 26, 53] have demonstrated tremendous success in various domains, assuming that the training and test data are independent and identically distributed. However, in many real-world scenarios, the training and testing datasets are often collected under different circumstances, leading to poor performance of DNNs trained on the source data when applied to out-of-distribution [18] target data. This performance decay, caused by domain shift, diminishes the generalization capability of DNNs. The domain generalization literature (DG) [5, 29, 31, 46] aims to overcome this challenge by leveraging the diversity of source domains to enhance model generalization.

In contrast to domain adaptation [51–53, 61] tasks that utilize both source and target domains, domain generalization tasks rely solely on source domains during training. As a result, prior research has focused primarily on developing methods to learn domain-invariant representations by aligning different source domains. However, contrastive learning has emerged as a promising solution to tackle this challenge. This approach involves creating positive and negative pairs and optimizing a distance metric that brings positive pairs closer together while pushing negative pairs apart. By optimizing the contrastive-based objective, the network can acquire generalized features that capture diverse sample-to-sample relationships across multiple domains. Recent studies have demonstrated the effectiveness of contrastive learning in domain generalization tasks [57, 62, 66].

After conducting to domain generalization, we have found that certain conventional contrastive-based methods [21, 23, 60] are not suitable for domain generalization. These methods use adversarial learning to learn domain-invariant features independently. However, it is typical for some categories, such as people and bikes, which frequently appear in the source domain, to be absent in the target domain. This leads to image-level bias, resulting in learned features being unstable and susceptible to erroneous generalization to the target domain. Although some existing works have achieved promising results by employing category centroids calculated from the source domain to facilitate generalization [30, 56], there are still

several limitations that need to be addressed. Firstly, while the centroid can represent the general appearance of a category, it may not fully encapsulate the full range of variations in certain attributes, such as color, texture, and lighting. As a result, this approach may decrease the diversity of categories, which can degrade the discriminative ability of the learned feature representations. Secondly, there has been no attempt made in this approach to quantify the distance between different category features. This can pose challenges in distinguishing categories with similar distributions in the target domain, especially in scenarios where no supervision information is available, leading to results with significant variance.

(a) cat (b) dog (c) fox

Figure 1: The fox can be seen as a mix between this cat and this dog.

Based on our analysis, we have adopted a new approach to domain generalization different from domain adaptation [30]. We reduce domain shift by learning sample-wise representations that attract similar samples and repel different samples. Firstly, we also use the distribution of each category in the source domain as a holistic representative to guide category alignment directions. This distribution can be accurately estimated with sufficient supervision from source data. Secondly, increasing intra-category compactness and inter-category separability of sample-wise representations can lead to better classification results. Thus, we separate sample-wise representations in the source domain and implicitly define an infinite number of positive pairs for each sample by sampling from the estimated distribution in the same category. This novel contrastive domain generalization approach can significantly enhance the model's capacity for generalization.

Meanwhile, many of the existing methods rely on batch normalization, which may not effectively learn domain-invariant representations. In our approach, we aim to learn domain-invariant representations by collecting domain-dependent batch normalization (BN) statistics for each available domain during training. As an example, human visual cognition is remarkable in its ability to link visual concepts to a combination of other concepts. For instance, when presented with images of a dog, a cat, and a fox, a child may naturally describe the fox as a cross between a cat and a dog (as illustrated in Figure 1). Furthermore, the child can understand much about the concept of a fox based on prior knowledge of what cats and dogs look like. Using a loose mathematical analogy, if visual representations of cats and dogs can be encapsulated as functions ϕ_{cat} and ϕ_{dog}, respectively, it should be possible to construct a representation for foxes $\phi_{fox} = f(\phi_{cat}, \phi_{dog}, \alpha)$, where f represents how the functions should be combined and is parameterized by α. Moreover, it should be easier to deduce the value of α than to determine ϕ_{fox} directly.

Taking inspiration from the normalization approach mentioned above, we investigate the possibility of applying similar techniques to domain generalization to improve learning efficiency. Recent studies, such as [10], have demonstrated the feasibility of training a single network to perform visual recognition across multiple domains. This is achieved by training the network to generate universal image representations, utilizing (i) convolutional kernels to extract domain-independent information from the world, and (ii) batch normalization (BN) layers to convert internal representations to the relevant target domains. Similarly, in the field of style transfer, Instance Normalization (IN) [42] has shown that a single network can be equipped with multiple distinct styles by encoding style information in the network's IN layers, allowing each style to be selectively applied to a target image. These findings provide evidence for the potential of normalization layers to encode transforms that can effectively represent visual concepts.

Building on our initial discussion, we put forth a hypothesis in this paper that normalization layers, such as BN, can be trained to differentiate specific visual classes. Consequently, we propose that combining these normalization layers and interpolating within them can facilitate efficient learning of new and unseen classes. This approach only requires the manipulation of normalization layers within the network, resulting in significantly fewer parameters to adjust than with full fine-tuning. This reduction in parameters also decreases the potential for overfitting, enabling training with smaller datasets. Our contributions can be summarized as follows:

(1) Our proposed method is contrastive domain generalization, a novel learning algorithm designed for domain generalization tasks. At its core, this approach aims to promote connections between sample-wise representations of the same category while simultaneously penalizing connections between sample-wise representations of different categories.
(2) We propose a novel interpolated normalization approach to generalize to new unseen classes, and we are the first to use interpolated batch normalization in contrastive learning to improve generalization performance.
(3) Through extensive testing, we have demonstrated that our method surpasses traditional normalization techniques and significantly enhances the performance of domain generalization.

2 RELATED WORK

Domain Generalization. The majority of domain generalization techniques focus on training models to adapt to domain shift by exposing them to various domains during the training phase. To promote invariance, multiple levels of encouragement can be employed:

"Feature-level" refers to methods that derive domain-invariant features by reducing the discrepancy between multiple training domains. Ghifary [13] introduced domain generalization to the deep learning community by training multi-task autoencoders that transform images from one source domain into various ones, thus learning invariant features. Similarly, Li [29] extended adversarial autoencoders by minimizing the Maximum Mean Discrepancy

measure to align the distributions of source domains to an arbitrary prior distribution through adversarial feature learning. Conditional Invariant Adversarial Networks [32] have been proposed to learn domain-invariant representations, while Deep Separation Networks [3] extract image representations divided into two subspaces: one unique to each domain and one shared. In contrast, Motiian [41] proposed to learn a discriminative embedding subspace through a Siamese architecture [25]. Episodic training [28] was developed to train a generic model while exposing it to domain shift. In each episode, a feature extractor is paired with a poorly tuned classifier to obtain robust features. Recently, Matsuura [40] proposed a method that simultaneously discovers latent domains by clustering features together and minimizing feature discrepancies between them. However, the limited variety of domains to which the model can be exposed during training can restrict the magnitude of the shift to which the model learns invariance.

"Data-level" methods are designed to alleviate training set domain bias by enriching the sample pool with a greater quantity and variety of data. To achieve this goal, researchers have proposed diverse data augmentation [47] techniques, including those based on domain-guided perturbations and adversarial examples, which aim to train models that can withstand distribution shift. Additionally, domain randomization [34, 48] has been employed to address the difficulty of transferring models from synthetic to real data, by supplementing synthetic data with random renderings. However, while these methods encourage the development of domain-invariant features, it is important to note that discarding domain-specific information may ultimately impair performance. Therefore, we assert that a balanced approach is necessary to achieve the best results.

"Model-based" methods are approaches that utilize customized architectures to address the domain generalization problem. For instance, [26] introduced a low-rank parameterized CNN model, which is a dynamically parameterized neural network that builds on the shallow binary undo bias method [22]. Similarly, in [11], a structured low-rank constraint is leveraged to align multiple domain-specific networks and a domain-invariant one. Mancini [36] trains multiple domain-specific classifiers and estimates the probabilities that a target sample belongs to each source domain to fuse the classifiers' predictions. Another recent work [5] proposes a different approach to address domain generalization by training a model to solve jigsaw puzzles while also performing well on a task of interest. However, most of these methods require changes to state-of-the-art architectures, resulting in an increased number of parameters or complexity of the network.

Batch Normalization To align the training distribution with the test distribution, the use of separate batch normalization statistics was first introduced in domain adaptation [6, 7, 33]. The same domain-dependent batchnorm layer has been adapted for the multi-domain scenario in [37, 39], and has been utilized in a graph-based approach [38] that leverages domain meta-data to improve the alignment of unknown domains with known ones. However, all of these methods require some form of target domain representation, such as samples or metadata, to perform alignment during training. Our method is inspired by recent advancements in contrastive learning and revised batch normalization techniques.

3 METHOD

3.1 Problem Formulation

The objective of Domain Generalization (DG) is to train a model that can generalize to previously-unseen target domains by utilizing multiple source domains. Both the source and target domains share a common label space denoted by $\mathcal{D} = \{D_1, D_2, \dots D_K\}$. In each domain, samples are drawn from a dataset $D_k = (x_i^k, y_i^k)_{i=1}^{N_t}$, where N_t represents the number of labeled samples in the domain D_k. The primary goal is to learn a generalized model G from a set of source datasets that can effectively perform on target data.

3.2 Review Contrastive-based Loss

In recent years, Contrastive learning [4, 9, 15, 16] has emerged as a highly effective method for learning meaningful representations from unlabeled data. The approach involves using an embedding function f, typically implemented using a convolutional neural network, to transform an input sample x into an embedding vector. The resulting vector z is then normalized onto a unit sphere. Pairs of similar samples are represented as (x, x^+), while dissimilar pairs are represented as (x, x^-). A popular contrastive loss function, such as InfoNCE [49], is used to train the model.

$$\mathbb{E}_{x,x^+,\{x^{n-}\}_{n=1}^N}\left[-\log\frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau}+\sum_{n=1}^N e^{f(x)^\top f(x^{n-})/\tau}}\right]. \quad (1)$$

To apply contrastive learning in practice, the expectation in the contrastive loss function is replaced with an empirical estimate. As demonstrated earlier, the contrastive loss is essentially a softmax formulation with a temperature parameter τ [49].

The contrastive loss introduced above intuitively promotes instance discrimination. However, inspired by the recent work on SCDA [30], which utilizes dense pixel predictions for domain adaptation in semantic segmentation and demonstrates that pixel-wise representation alignment outperforms existing algorithms by a significant margin, we explore the use of contrastive learning from the perspective of semantic data augmentation in domain generalization to enhance model performance.

3.3 Semantic Distribution Estimation

To enhance source feature $\boldsymbol{a}_i$, we establish a multivariate normal distribution $\mathcal{N}(0, \Sigma_{y_i})$ with zero mean. The covariance matrix Σy_i depends on the features of all samples in class y_i, and is computed online by combining statistics from all mini-batches during implementation. Formally, the algorithm for estimating online covariance matrices is represented as:

$$\boldsymbol{\mu}_j^{(t)} = \frac{n_j^{(t-1)}\boldsymbol{\mu}_j^{(t-1)} + m_j^{(t)}\boldsymbol{\mu'}_j^{(t)}}{n_j^{(t-1)} + m_j^{(t)}}, \quad (2)$$

$$\begin{aligned}\Sigma_j^{(t)} =& \frac{n_j^{(t-1)}\Sigma_j^{(t-1)} + m_j^{(t)}\Sigma'_j{}^{(t)}}{n_j^{(t-1)} + m_j^{(t)}} \\ &+ \frac{n_j^{(t-1)} m_j^{(t)}(\boldsymbol{\mu}_j^{(t-1)} - \boldsymbol{\mu'}_j^{(t)})(\boldsymbol{\mu}_j^{(t-1)} - \boldsymbol{\mu'}_j^{(t)})^{\mathrm{T}}}{(n_j^{(t-1)} + m_j^{(t)})^2},\end{aligned} \quad (3)$$

$$n_j^{(t)} = n_j^{(t-1)} + m_j^{(t)}, \quad (4)$$

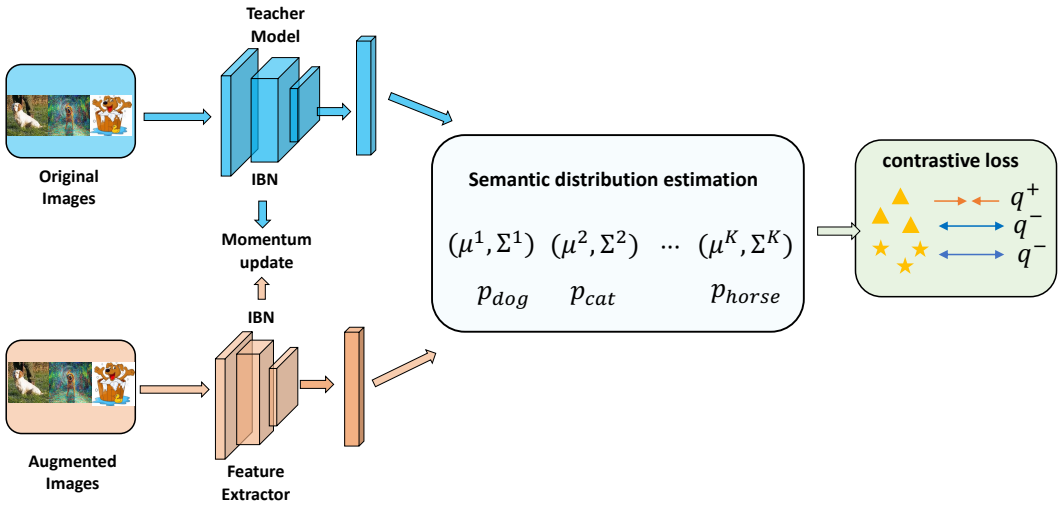

Figure 2: The framework of INC. Our framework contains two key components, namely interpolation normalization and contrast domain generalization.

At the t^{th} step, ${\mu'}_j^{(t)}$ and ${\Sigma'}_j^{(t)}$ are utilized as the estimates of the average values and covariance matrices of the features associated with the j^{th} class. The average values and covariance matrices of the features belonging to the j^{th} class in the t^{th} mini-batch are represented by ${\mu'}_j^{(t)}$ and ${\Sigma'}_j^{(t)}$, respectively. $n_j^{(t)}$ refers to the total number of training samples that belong to the j^{th} class across all t mini-batches, whereas $m_j^{(t)}$ signifies the number of training samples that belong to the j^{th} class exclusively in the t^{th} mini-batch. Throughout the generalization process, we continually update the semantic distributions for each source training image on a dynamic basis. The estimated semantic distributions are particularly useful for facilitating category alignment with a higher degree of accuracy. We leverage sample-wise contrastive loss as :

$$\mathcal{L}_i^{M,N} = -\frac{1}{M}\sum_{m=1}^{M}\log\frac{e^{q_i^\top q^{m+}/\tau}}{e^{q_i^\top q^{m+}/\tau}+\sum_{j=1}^{K-1}\frac{1}{N}\sum_{n=1}^{N}e^{q_i^\top q_j^{n-}/\tau}}, \tag{5}$$

where q^{m+} denotes the m^{th} positive example from the identical category as q_i, while q_j^{n-} represents the n^{th} negative example from a distinct category, indexed by j. We take an infinity limit on the number of M and N, where the effect of M and N is hopefully absorbed in a probabilistic way. With this application of infinity limit, the statistics of the data are sufficient to achieve the same goal of multiple pairing. Mathematically, as M and N goes to infinity, $\mathcal{L}^{M,N}$ becomes the estimation of:

$$\begin{aligned}\mathcal{L}_i^{\infty} &= \lim_{\substack{M\to\infty\\N\to\infty}}\mathcal{L}_i^{M,N}\\ &= -\mathbb{E}_{\substack{q^+\sim p(q^+)\\ q_j^-\sim p(q_j^-)}}\log\frac{e^{q_i^\top q^+/\tau}}{e^{q_i^\top q^+/\tau}+\sum_{j=1}^{K-1}e^{q_i^\top q_j^-/\tau}},\end{aligned} \tag{6}$$

where $p(q^+)$ is the positive semantic distribution that has the same semantic label and $p(q_j^-)$ is the j^{th} negative semantic distribution that has different semantic label with respect to q_i. According to SCDA, the upper bound loss can be written as follows:

$$\begin{aligned}&-\mathbb{E}_{q^+,q_j^-}\log\frac{e^{q_i^\top q^+/\tau}}{e^{q_i^\top q^+/\tau}+\sum_{j=1}^{K-1}e^{q_i^\top q_j^-/\tau}}\\ &=\mathbb{E}_{q^+}\left[\log\left[e^{\frac{q_i^\top q^+}{\tau}}+\sum_{j=1}^{K-1}\mathbb{E}_{q_j^-}e^{\frac{q_i^\top q_j^-}{\tau}}\right]\right]-\mathbb{E}_{q^+}\left[\frac{q_i^\top q^+}{\tau}\right] &(7)\\ &\le\log\left[\mathbb{E}_{q^+}\left[e^{\frac{q_i^\top q^+}{\tau}}+\sum_{j=1}^{K-1}\mathbb{E}_{q_j^-}e^{\frac{q_i^\top q_j^-}{\tau}}\right]\right]-q_i^\top\mathbb{E}_{q^+}\left[\frac{q^+}{\tau}\right] &(8)\\ &=\log\left[\mathbb{E}_{q^+}e^{\frac{q_i^\top q^+}{\tau}}+\sum_{j=1}^{K-1}\mathbb{E}_{q_j^-}e^{\frac{q_i^\top q_j^-}{\tau}}\right]-q_i^\top\mathbb{E}_{q^+}\left[\frac{q^+}{\tau}\right] &(9)\\ &=\mathcal{L}_c &(10)\end{aligned}$$

Note that the contrastive loss is employed in all source domains simultaneously. As contrastive loss only involves modifying the fully connected layer and ignores changes to the batch normalization layer, in order to further enhance domain generalization performance, we consider modifying the batch normalization layer. We first review batch normalization, and then propose a method to interpolate within the learning batch normalization layer to effectively learn new classes.

3.4 Vanilla Batch Normalization

To begin, we will provide a brief overview of the batch normalization (BN) transform [20]. Consider the activations x_i of a single example i within a mini-batch of size m. The BN transform can be expressed as:

$$\text{BN}(x_i) = \gamma\widehat{x}_i + \beta. \tag{11}$$

Given the mean $\mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m}x_i$, variance $\sigma_{\mathcal{B}}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu_{\mathcal{B}})^2$, and the normalized input $\widehat{x}_i =$

$(x_i - \mu_{\mathcal{B}}) / \sqrt{\sigma^2_{\mathcal{B}} + \epsilon}$. The parameters γ and β are learnable, while ϵ is a small positive constant used to avoid division by zero.

In the following sections, we will demonstrate the construction of component BNs and elaborate on two methods of integrating them to interpolate novel classes in domain generalization.

3.5 Component Generation

To construct component BNs that are specifically designed to discriminate a particular object class, a simple approach is to extract BNs from a network that has been trained on a corresponding classification task. This involves using a pre-trained network, referred to as the template network, as a starting point. To generate BN layers that can detect a concept such as cats, for example, we fine-tune the network on a dataset that contains examples of cats and non-cats. During this process, we adjust only the BN and last classification layer parameters. The resulting fine-tuned network becomes a component network that is capable of detecting cats. We repeat this process for other object classes, always starting from the same template network, until we have obtained the desired number of component networks.

3.6 Interpolating Component Networks

Once the BN components have been computed and selected for a particular novel target class, we utilize two methods for interpolation:

1. Composite Batch Normalization (ComBN), providing a linear combination of generated BN components.

2. Principal Component Batch Normalization (PCBN), providing a PCA-based latent space interpolation.

Composite Batch Normalization (ComBN). We leverage the ComBN transform as a linear combination of the generated BN components, using a notation similar to the one mentioned above.

$$\mathrm{ComBN}_{\alpha}\left(x_i\right) = \sum_{j=1}^{J} \alpha_j \mathrm{BN}_j\left(x_i\right). \tag{12}$$

The ComBN transform is represented as follows, where J is the number of BN components that make up a ComBN, and $alpha_j$ are learnable scalar coefficients that represent the interpolation weights. These coefficients are initialized to $1/J$.

In practice, once the component networks have been generated, we replace each BN layer in the original template network with a ComBN. This ComBN is constructed from BN layers of selected component networks that originate from the same depth-wise layer position. We then train the ComBN network by optimizing α_j and the last layer to the target task using standard techniques.

It is worth noting that the component BNs in the ComBN network are always used in inference mode. This means that their γ, β, running mean, and running variance are fixed, and the running mean and variance are used instead of the mini-batch mean μ and variance $\sigma_{\mathcal{B}}$ when evaluating Eq. 11. Moreover, this formulation often leads to a significant reduction in the number of parameters, which can be advantageous in mitigating overfitting when the training data is limited.

Principal Component Batch Normalization. Another approach to utilizing the information contained in BN components is to first use them to learn a mapping for their parameters in a latent space, and then perform optimization in this space.

To implement this using PCA, we begin by stacking the row vectors of γ andβ parameters that originate from each BN component j to form $J \times C$ matrices Γ and B, respectively. Here, J is the number of components and C is the number of channels in each component BN layer. We then mean-center Γ and B by subtracting their column-wise mean vectors μ_γ and μ_β, resulting in $\mathbf{X}_\gamma = \mathbf{\Gamma} - \mu_\gamma$ and $\mathbf{X}_\beta = \mathbf{B} - \mu_\beta$. Next, we apply singular value decomposition to obtain principal axes matrices $\mathbf{V}_\gamma^\top$ and $\mathbf{V}_\beta^\top$.

$$\mathbf{U}_\gamma \mathbf{S}_\gamma \mathbf{V}_\gamma^\top \leftarrow \mathbf{X}_\gamma, \tag{13}$$

$$\mathbf{U}_\beta \mathbf{S}_\beta \mathbf{V}_\beta^\top \leftarrow \mathbf{X}_\beta \tag{14}$$

We set the number of dimensions in our latent space to the maximum possible, which is $min(J, C)$. We then train latent space parameter vectors g and b, which are initialized by transforming the existing BN weights of the template network to the latent space. Finally, we transform these vectors back to the parameter space using the principal axes matrices.

$$\gamma = g\mathbf{V}_\gamma^\top + \mu_\gamma, \tag{15}$$

$$\beta = b\mathbf{V}_\beta^\top + \mu_\beta \tag{16}$$

We then apply this approach in a similar manner to ComBN by replacing the BN layers in the original template with PCBN. Essentially, this is like using standard BN, except for the optimization of parameters in the latent space. Unlike ComBN, where interpolation is directly performed in the parameter space of the original component class (in the form of frozen BN components), here we first attempt to distill the concepts of class into principal classes in the latent space before optimizing them.

3.7 Final Objective

With the contrast domain generalization and interpolating normalization, we can define our final objective for effective domain generalization (as illustrated in Figure. 2). First, a recent work of FACT [58], which leverages fourier co-teacher loss (CTL) and co-teacher regularization (CTR). Combining all these losses function together, we can get the objective of FACT as:

$$\mathcal{L}_{FACT} = \mathcal{L}_{CTL} + \alpha_1 \mathcal{L}_{CTR} \tag{17}$$

where α_1 is constant controlling the strength of corresponding loss. Further, We employ the contrast domain generalization loss and interpolating normalization to FACT to enhancing the intra-class compactness and inter-class separability in a unified framework, and the total loss can be written as follows:

$$\mathcal{L}_{INC} = \mathcal{L}_{CTL} + \alpha_1 \mathcal{L}_{CTR} + \alpha_2 \mathcal{L}_c \tag{18}$$

where α_2 controls the trade-off between FACT loss and contrast domain generalization loss.

Table 1: Leave-one-domain-out results on Digits-DG. The best is bolded.

Methods	MNIST	MNIST-M	SVHN	SYN	Avg.
DeepAll [64]	95.8	58.8	61.7	78.6	73.7
Jigen [5]	96.5	61.4	63.7	74.0	73.9
CCSA [41]	95.2	58.2	65.5	79.1	74.5
MMD-AAE [29]	96.5	58.4	65.0	78.4	74.6
CrossGrad [47]	96.7	61.1	65.3	80.2	75.8
DDAIG [65]	96.6	64.1	68.6	81.0	77.6
L2A-OT [64]	96.7	63.9	68.6	83.2	78.1
FACT [58]	97.9	65.6	72.4	90.3	81.5
INC (*ours*)	**98.3**	**69.2**	**74.7**	**92.5**	**83.7**

4 EXPERIMENTS

4.1 Datasets

We conduct experiments on four commonly utilized domain generalization (DG) datasets, namely Digits-DG [64], PACS [26], Office-Home [50], and DomainNet [44]. Below are brief descriptions of each dataset:

Digits-DG [64] dataset consists of four digit domains: MNIST, MNIST-M, SVHN, and SYN, which exhibit significant variations in font style, background, and stroke color. We randomly choose 600 images per class for each domain and divide the data into 80% for training and 20% for validation.

PACS [26] dataset is specifically designed for DG and comprises 9,991 images from four domains (Art, Cartoon, Photo, and Sketch) with significant style differences. Each domain includes seven categories: dog, elephant, giraffe, guitar, house, horse, and person. To ensure a fair comparison, we utilize the original training-validation split provided by the dataset.

Office-Home [50] dataset is an object recognition dataset that comprises 15,500 images of 65 categories in office and home environments. These 65 categories are shared by four domains (Art, Clipart, Product, and Real-World), which exhibit variations in viewpoint and image style. We divide each domain into 90% for training and 10% for validation.

DomainNet [44] dataset is a vast collection of 586,575 images of 345 classes from six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. To ensure fair comparisons with previous studies, we adopt the leave-one-domain-out protocol. Specifically, we select one domain as the test domain and utilize the remaining domains as the source domains. The model that achieves the best performance on the validation splits of all source domains is selected as the final model. The top-1 classification accuracy is used as the evaluation metric.

4.2 Implementation details

We closely follow the implementations of [58]. Here we briefly introduce the main details for training our model.

For Digits-DG, we use the same backbone network as [58]. We train the network from scratch using SGD, batch size of 128 and weight decay of $5e-4$ for 50 epochs. The initial learning rate is set to 0.05 and and decayed by 0.1 every 20 epochs. For PACS,Office-Home and DomainNet, we use the ImageNet pretrained ResNet [17] as our backbone. We train the network with SGD, batch size of 16

Table 2: Leave-one-domain-out results on PACS with ResNet-18. The best is bolded.

Methods	Art	Cartoon	Photo	Sketch	Avg.
DeepAll[64]	77.63	76.77	95.85	69.50	79.94
MetaReg [2]	83.70	77.20	95.50	70.30	81.70
JiGen [5]	79.42	75.25	96.03	71.35	80.51
Epi-FCR [28]	82.10	77.00	93.90	73.00	81.50
MMLD [40]	81.28	77.16	96.09	72.29	81.83
DDAIG [65]	84.20	78.10	95.30	74.70	83.10
CSD [45]	78.90	75.80	94.10	76.70	81.40
MASF [12]	80.29	77.17	94.99	71.69	81.04
L2A-OT [64]	83.30	78.20	96.20	73.60	82.80
EISNet [55]	81.89	76.44	95.93	74.33	82.15
FACT [58]	85.90	79.35	96.61	80.88	85.69
INC (*ours*)	**87.91**	**82.37**	**97.85**	**84.64**	**88.19**

Table 3: Leave-one-domain-out results on PACS with ResNet-50. The best is bolded.

Methods	Art	Cartoon	Photo	Sketch	Avg.
DeepAll[64]	84.94	76.98	97.64	76.75	84.08
MetaReg [2]	87.20	79.20	97.60	70.30	83.60
MASF [12]	82.89	80.49	95.01	72.29	82.67
EISNet [55]	86.64	81.53	97.11	78.07	85.84
RSC [19]	87.89	82.16	97.92	83.35	87.83
ATSRL [59]	90.0	83.5	98.9	80.0	88.1
MDGH [35]	86.7	82.3	98.4	82.7	87.5
FACT [58]	90.89	83.65	97.78	86.17	89.62
INC (*ours*)	**91.37**	**85.48**	**98.33**	**88.21**	**90.85**

Table 4: Leave-one-domain-out results on Office-Home. The best is bolded.

Methods	Art	Clipart	Product	Real	Avg.
DeepAll [64]	57.88	52.72	73.50	74.80	64.72
CCSA [41]	59.90	49.90	74.10	75.70	64.90
MMD-AAE [29]	56.50	47.30	72.10	74.80	62.70
CrossGrad [47]	58.40	49.40	73.90	75.80	64.40
DDAIG [65]	59.20	52.30	74.60	76.00	65.50
L2A-OT [64]	60.60	50.10	74.80	77.00	65.60
Jigen [5]	53.04	47.51	71.47	72.79	61.20
RSC [19]	58.42	47.90	71.63	74.54	63.12
FACT [58]	60.34	54.85	74.48	76.55	66.56
INC (*ours*)	**61.37**	**56.79**	**76.23**	**77.38**	**67.94**

Table 5: Leave-one-domain-out results on DomainNet. The best is bolded

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
C-DANN [32]	54.60	17.30	43.70	12.10	56.20	45.90	38.30
RSC [19]	55.00	18.30	44.40	12.20	55.70	47.80	38.90
Mixup [63]	55.70	18.50	44.30	12.50	55.80	48.20	39.20
SagNet [43]	57.70	19.00	45.30	12.70	58.10	48.80	40.30
MLDG [27]	59.10	19.10	45.80	13.40	59.60	50.20	41.20
ERM [14]	58.10	18.80	46.70	12.20	59.60	49.8	40.90
MetaReg [2]	59.77	25.58	50.19	11.52	64.56	50.09	43.62
DMG [8]	65.24	22.15	50.03	15.68	59.63	49.02	43.63
SelfReg [23]	62.40	22.60	51.80	14.30	62.50	53.80	44.60
SADML [54]	63.72	23.36	51.92	15.93	63.01	54.34	45.38
INC (*ours*)	**64.58**	**24.47**	**52.38**	**16.71**	**64.45**	**56.70**	**46.55**

and weight decay of $5e-4$ for 50 epochs. The initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. Images fed into the networks are using interpolation and normalized. Our method is implemented with the PyTorch library on Nvidia Tesla V100.

4.3 Quantitative results in image classification

Evaluation on Digits-DG. The results of our evaluation on Digits-DG are presented in Table 1. Our method outperforms all competitors, with an average improvement of over 2% compared to the baseline FACT citexu2021fourier. Notably, our method achieves particularly strong results on the challenging target domains of SVHN and SYN, which feature cluttered digits and low image quality. In these domains, our method outperforms FACT by a significant margin of 2.3% and 2.2%, respectively. These results demonstrate that our approach is effective in clustering sample representations from the same category and learning domain-invariant features during model training

Evaluation on PACS. We present the quantitative results of our method and existing approaches in Table. 2 and Table. 3, using the FACT as the baseline model. Our method consistently achieves the highest average accuracy across all backbone networks. Specifically, our approach outperforms existing methods in three test domains (Art, Cartoon, and Sketch) using both ResNet-18 and ResNet-50. Notably, our method clearly surpasses MixStyle, a style augmentation technique based on linear interpolation of known styles, when incorporating ResNet-18. Unlike previous methods that require data generators or domain labels to synthesize novel domain samples, such as L2A-OT and DDAIG, our approach does not rely on either of these. These results demonstrate the effectiveness of our method for domain generalization and support our approach of contrast domain generalization and interpolation normalization.

Evaluation on Office-Home. The Office-Home dataset consists of four domains with relatively low domain discrepancy compared to other datasets. As shown in Table. 4, despite the small domain gap in this benchmark, which makes it challenging to generalize to novel styles, our method achieves comparable results to the state of the art. Notably, our approach consistently improves the performance of the baseline model in all domains, while many existing methods struggle in certain domains. Additionally, our method outperforms approaches that synthesize novel domain samples in terms of average accuracy.

Table 6: Ablation study on different components of our method on the PACS datasets (ResNet-18).

Methods	Art	Cartoon	Photo	Sketch	Avg.
FACT	85.90	79.35	96.61	80.88	85.69
FACT + (CDG)	86.13	81.34	97.14	82.33	86.74
FACT + (IBN)	86.54	81.23	96.99	83.18	87.00
INC (*ours*)	**87.91**	**82.37**	**97.85**	**84.64**	**88.19**

Evalution on DomainNet Table. 5 displays the results on DomainNet, which consists of six domains with significantly larger domain discrepancy than other datasets. On this more challenging benchmark, our method achieves better average accuracy than existing approaches and improves the top-1 averaged accuracy by 1.17%. Notably, our method outperforms the baseline model, while many existing methods perform worse than the baseline when using ResNet-50 as the backbone network.

4.4 Experimentai Analysis

Impact of different components. We conduct an extensive ablation study to investigate the role of each component in our method INC in Table. 6. FACT denotes that we only use FACT without any other strategy. Based on FACT, we add a CDG (contrast domain generalization) to obtain FACT+(CDG) and IBN (interpolation normalization) to obtain FACT + (IBN). We can see that our method can improve the baseline FACT by a large margin. This confirm the effectiveness of revised contrastive learning and interpolation BN in learning domain-invariant features and improving diversity.

Compare on different contrastive-based loss. To demonstrate the effectiveness of the proposed contrast domain generalization, we compare it with other classical contrastive-based losses such as proxy-based loss [60] and proxy-anchor loss [24]. As shown in Table. 8, we can find that our method surpasses both proxy-based methods and contrastive-based methods.

Evaluation of domain divergence. We employ the widely used $\mathcal{A}$-distance metric to assess the distribution divergence by measuring the dissimilarity between domains. This metric is derived from the classification error of a trained classifier that differentiates the source and target domains, and is defined as $\mathcal{A}_{dis} = 2(1-\epsilon)$. A

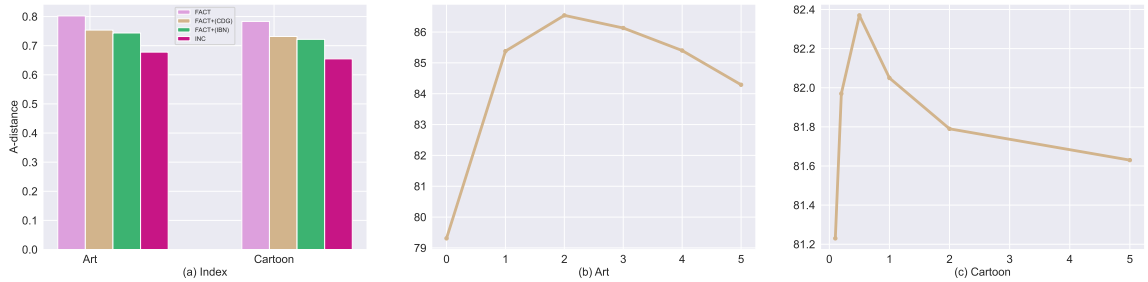

Figure 3: (a) $\mathcal{A}$-distance between different method including FACT, FACT+(CDG), FACT+(IBN), INC. (b) hyper-parameter α_1 on the task Art. (c) hyper-parameter α_2 on the task Cartoon.

Table 7: We present the outcomes of PACS' single source domain generalization. The rows and columns indicate the source and target domains, respectively. The accuracy, along with the absolute gain from the baseline in brackets, is reported. Any positive gain is displayed in green.

	Photo	Art	Cartoon	Sketch
Photo	99.78 (+0.00)	65.27 (+1.25)	24.38 (+1.97)	54.39 (+4.31)
Art	98.37 (+1.31)	99.37 (+0.00)	69.56 (+1.83)	53.87 (+1.25)
Cartoon	86.38 (+1.85)	72.38 (+1.37)	99.35 (+0.00)	72.87 (+1.96)
Sketch	43.87 (+1.15)	45.78 (+1.49)	62.83 (+1.26)	99.83 (+0.00)

Table 8: Ablation study on different contrastive-based loss on Office-Home on ResNet-18.

loss function	Avg.
proxy-based loss [60]	68.92
proxy-anchor loss [24]	62.48
INC (*ours*)	**71.34**

smaller $\mathcal{A}$-distance indicates better concurrence between the distributions. As illustrated in Figure. 3 (a), our approach concentrates on acquiring more invariant features, resulting in a decrease in the gap between the source and target domains in comparison to other methods.

Parameter Sensitivity. Figure. 3 show the sensitivity of MoEL to hyper-parameter α_1 and α_2. Specifically, the value of α_1 varies from $\{0, 1, 2, 3, 4, 5\}$, α_2 varies from $\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$ It can be observed that MoEL achieves the best performance when $\alpha_1 = 2$ and $\alpha_2 = 0.5$, which further verify the stability of our method. We recommend $\alpha_1 = 2$ and $\alpha_2 = 0.5$ for a naive implement or a start point of hyper-parameter searching.

4.5 Single-Source Domain Generalization

We conducted an extreme case evaluation of our model for the domain generalization task. Unlike our previous experimental settings, we trained our model using examples from a single source domain and then assessed its performance using examples from other target domains. All source-target combinations were evaluated and the results are presented in Table. 7. To establish a baseline, we compared our model with FACT's performance under the same conditions (see scores in left and right tables) and highlighted any differences in the last row ('+' indicates our model performed better). The results in Table. 7 show that our model generally outperforms the alternative, with a significant improvement in average accuracy. Moreover, for the difficult task such "Photo" to "Sketch", he average accuracy increase by 4.31%, confirming the effectiveness of our method on challenging single-source tasks.

5 CONCLUSION

In this paper, we propose a new contrastive learning technique that encourages the extraction of class-discriminative and class-balanced features from source domains. The method promotes the clustering of sample representations from the same category, while dispersing those from different categories, thereby enhancing the model's generalization capacity. Additionally, most current contrastive learning methods employ batch normalization, which may hinder the learning of domain-invariant features. Drawing inspiration from recent investigations on universal representations for neural networks, we suggest a straightforward emulation of this mechanism by utilizing batch normalization layers to distinguish visual classes and devising a way to combine them for domain generalization tasks. Our experiments reveal a significant improvement in classification accuracy over state-of-the-art techniques on prevalent domain generalization benchmarks, such as Digits-DG, PACS, Office-Home, and DomainNet. We hope that our method can inspire other algorithms in the future.

6 ACKNOWLEDGE

This work was supported by National Natural Science Foundation (No.62102265), Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (No.GML-KF-22-29), Natural Science Foundation of Guangdong Province of China (No.2022A1515011474).

REFERENCES

[1] Yang Bai and Weiqiang Wang. 2019. Acpnet: anchor-center based person network for human pose estimation and instance segmentation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1072–1077.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. *NeurIPS* 31 (2018).

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in neural information processing systems*. 343–351.

[4] Qi Cai, Yu Wang, Yingwei Pan, Ting Yao, and Tao Mei. 2020. Joint Contrastive Learning with Infinite Possibilities. In *Proc. NeurIPS*.

[5] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain generalization by solving jigsaw puzzles. In *CVPR*. 2229–2238.

[6] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. 2017. Autodial: Automatic domain alignment layers. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 5077–5085.

[7] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. 2017. Just dial: Domain alignment layers for unsupervised domain adaptation. In *International Conference on Image Analysis and Processing*. Springer, 357–369.

[8] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. 2020. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*. 301–318.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. ICML*, Vol. 119. 1597–1607.

[10] Gratianus Wesley Putra Data, Kirjon Ngu, David William Murray, and Victor Adrian Prisacariu. 2018. Interpolating convolutional neural networks using batch normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 574–588.

[11] Zhengming Ding and Yun Fu. 2017. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing* 27, 1 (2017), 304–313.

[12] 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*. 6447–6458.

[13] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. 2015. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*. 2551–2559.

[14] Ishaan Gulrajani and David Lopez-Paz. 2021. In search of lost domain generalization. *ICLR* (2021).

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *Proc. CVPR*. 1735–1742.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. CVPR*. 9726–9735.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[18] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10951–10960.

[19] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In *ECCV*. 124–140.

[20] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*. 448–456.

[21] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. 2021. Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia*. 22–31.

[22] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *European Conference on Computer Vision*. Springer, 158–171.

[23] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. 2021. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ICCV*. 9619–9628.

[24] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. 2020. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3238–3247.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille.

[26] 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 (ICCV)*. 5542–5550.

[27] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2018. Learning to generalize: Meta-learning for domain generalization. In *AAAI*.

[28] 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.

[29] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018. Domain generalization with adversarial feature learning. In *CVPR*. 5400–5409.

[30] Shuang Li, Binhui Xie, Bin Zang, Chi Harold Liu, Xinjing Cheng, Ruigang Yang, and Guoren Wang. 2021. Semantic distribution-aware contrastive adaptation for semantic segmentation. *arXiv preprint arXiv:2105.05013* (2021).

[31] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. 2018. Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[32] 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 *ECCV*. 624–639.

[33] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. 2018. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition* 80 (2018), 109–117.

[34] Antonio Loquercio, Elia Kaufmann, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. 2019. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics* (2019).

[35] Divyat Mahajan, Shruti Tople, and Amit Sharma. 2021. Domain generalization using causal matching. In *ICML*. 7313–7324.

[36] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. 2018. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 1353–1357.

[37] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. 2018. Robust place categorization with deep domain generalization. *IEEE Robotics and Automation Letters* 3, 3 (2018), 2093–2100.

[38] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. 2019. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6568–6577.

[39] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. 2018. Boosting domain adaptation by discovering latent domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3771–3780.

[40] Toshihiko Matsuura and Tatsuya Harada. 2020. Domain generalization using a mixture of multiple latent domains. In *AAAI*. 11749–11756.

[41] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. 2017. Unified deep supervised domain adaptation and generalization. In *ICCV*. 5715–5725.

[42] Hyeonseob Nam and Hyo-Eun Kim. 2018. Batch-instance normalization for adaptively style-invariant neural networks. *Advances in Neural Information Processing Systems* 31 (2018).

[43] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. 2021. Reducing domain gap by reducing style bias. In *CVPR*. 8690–8699.

[44] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *ICCV*. 1406–1415.

[45] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. 2020. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*. 7728–7738.

[46] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2019. Multi-component image translation for deep domain generalization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 579–588.

[47] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. 2018. Generalizing across domains via cross-gradient training. *ICLR* (2018).

[48] 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.

[49] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).

[50] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*. 5018–5027.

[51] Mengzhu Wang, Paul Li, Li Shen, Ye Wang, Shanshan Wang, Wei Wang, Xiang Zhang, Junyang Chen, and Zhigang Luo. 2022. Informative pairs mining based adaptive metric learning for adversarial domain adaptation. *Neural Networks* 151 (2022), 238–249.

[52] Mengzhu Wang, Shanshan Wang, Wei Wang, Li Shen, Xiang Zhang, Long Lan, and Zhigang Luo. 2023. Reducing bi-level feature redundancy for unsupervised domain adaptation. *Pattern Recognition* 137 (2023), 109319.

[53] Mengzhu Wang, Wei Wang, Baopu Li, Xiang Zhang, Long Lan, Huibin Tan, Tianyi Liang, Wei Yu, and Zhigang Luo. 2021. Interbn: Channel fusion for adversarial

unsupervised domain adaptation. In *Proceedings of the 29th ACM international conference on multimedia*. 3691–3700.

[54] Mengzhu Wang, Jianlong Yuan, Qi Qian, Zhibin Wang, and Hao Li. 2022. Implicit Semantic Augmentation for Distance Metric Learning in Domain Generalization. *ACMMM* (2022).

[55] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. 2020. Learning from extrinsic and intrinsic supervisions for domain generalization. In *ECCV*. 159–176.

[56] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. 2023. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[57] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. 2021. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems* 34 (2021), 30414–30425.

[58] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. A fourier-based framework for domain generalization. In *CVPR*. 14383–14392.

[59] Fu-En Yang, Yuan-Chia Cheng, Zu-Yun Shiau, and Yu-Chiang Frank Wang. 2021. Adversarial Teacher-Student Representation Learning for Domain Generalization. *NeurIPS* 34 (2021).

[60] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. 2022. PCL: Proxy-based Contrastive Learning for Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7097–7107.

[61] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2019. Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2720–2729.

[62] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

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

[64] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 13025–13032.

[65] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Learning to generate novel domains for domain generalization. In *ECCV*. 561–578.

[66] Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. 2021. Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10306–10315.