SelfReg: Self-supervised Contrastive Regularization for Domain Generalization

Daehee Kim[1,2], Youngjun Yoo[1], Seunghyun Park[2], Jinkyu Kim[3*], and Jaekoo Lee[1*]
[1] College of Computer Science, Kookmin University [2] Clova AI Research, NAVER Corp.
[3] Department of Computer Science and Engineering, Korea University

Abstract

In general, an experimental environment for deep learning assumes that the training and the test dataset are sampled from the same distribution. However, in real-world situations, a difference in the distribution between two datasets, i.e. domain shift, may occur, which becomes a major factor impeding the generalization performance of the model. The research field to solve this problem is called domain generalization, and it alleviates the domain shift problem by extracting domain-invariant features explicitly or implicitly. In recent studies, contrastive learning-based domain generalization approaches have been proposed and achieved high performance. These approaches require sampling of the negative data pair. However, the performance of contrastive learning fundamentally depends on quality and quantity of negative data pairs. To address this issue, we propose a new regularization method for domain generalization based on contrastive learning, called self-supervised contrastive regularization (SelfReg). The proposed approach use only positive data pairs, thus it resolves various problems caused by negative pair sampling. Moreover, we propose a class-specific domain perturbation layer (CDPL), which makes it possible to effectively apply mixup augmentation even when only positive data pairs are used. The experimental results show that the techniques incorporated by SelfReg contributed to the performance in a compatible manner. In the recent benchmark, DomainBed, the proposed method shows comparable performance to the conventional state-of-the-art alternatives.

1. Introduction

Machine learning systems often fail to generalize out-of-sample distribution as they assume that in-samples and out-of-samples are independent and identically distributed – this assumption rarely holds during deployment in real-world scenarios where the data is highly likely to change over time and space. Deep convolutional neural network features are often domain-invariant to low-level visual cues [35], some studies [10] suggest that they are still susceptible to domain shift.

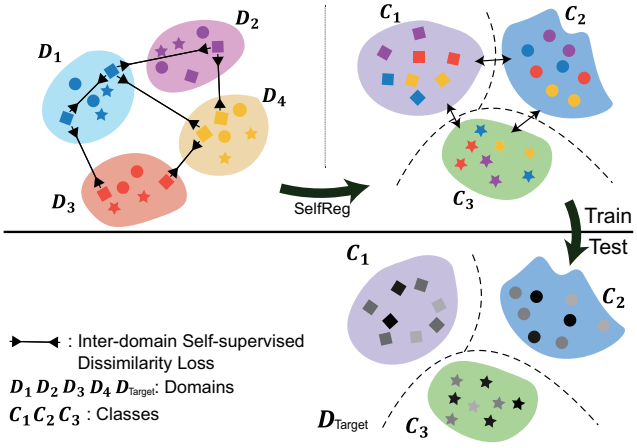

Figure 1. Our model utilizes the self-supervised contrastive losses for the model to learn domain-invariant representation by mapping the latent representation of the same-class samples close together. Note that different shapes (i.e. circles, stars, and squares) indicate different classes $C_{i\in\{1,2,3\}}$, and we differently color-code according to their domain $\mathcal{D}_{i\in\{1,2,3,4,\text{Target}\}}$.

There have been increasing efforts to develop models that can generalize well to out-of-distribution. The literature in domain generalization (DG) aims to learn the invariances across multiple different domains so that a classifier can robustly leverage such invariances in unseen test domains [40, 15, 29, 28, 31, 38]. In the domain generalization task, it is assumed that multiple source domains are accessible during training, but the target domains are not [4, 31]. This is different from domain adaptation (DA), semi-supervised domain adaptation (SSDA), and unsupervised domain adaptation (UDA) problems, where examples from the target domain are available during training. In this paper, we focus on the domain generalization task.

Some recent studies [7, 20, 33] suggest that contrastive learning can be successfully used in a self-supervised learning task by mapping the latent representations of the posi-

*Corresponding authors: J. Kim (jinkyukim@korea.ac.kr) and J. Lee (jaekoo@kookmin.ac.kr)

tive pair samples close together, while that of negative pair samples further away in the embedding space. Such a contrastive learning strategy has also been utilized for the domain generalization tasks [30, 11], similarly aiming to reduce the distance of same-class features in the embedding space, while increasing the distance of different-class features. However, such negative pairs often make the training unstable unless useful negative samples are available in the same batch, which is but often challenging.

In this work, we revisit contrastive learning for the domain generalization task, but only with positive pair samples. As it is generally known that using positive pair samples only causes the performance drop, which is often called representation collapse [17]. Inspired by recent studies on self-supervised learning [8, 17], which successfully avoids representation collapse by placing one more projection layer at the end of the network, we successfully learn domain-invariant features and our model trained with self-supervised contrastive losses shows the matched or better performance against alternative state-of-the-art methods, where ours is ranked at top places in the domain generalization benchmarks, i.e. DomainBed [18].

However, self-supervised contrastive losses are only part of the story. As we generally use a linear form of the loss function, properly balancing gradients is required so that network parameters converge to generate domain-invariant features. To mitigate this issue, we advocate for applying the following three gradient stabilization techniques: (i) loss clipping, (ii) stochastic weights averaging (SWA), and (iii) inter-domain curriculum learning (IDCL). We observe that the combined use of these techniques further improves the model's generalization power.

To effectively evaluate our proposed model, we first use the publicly available domain generalization data set called PACS [26], where we analyzed our model in detail to support our claims. We further experiment with much larger benchmarks called DomainBed [18] where our model shows matched or better performance against alternative state-of-the-art methods.

We summarize our main contributions as follows:

- SelfReg facilitates the application of metric learning using only positive pairs without negative pairs.
- We devised a CDPL by exploiting a condition that use only positive pairs. The combination of CDPL and mixup improves the weakness of mixup approach.
- The performance comparable to that of the SOTA DG methods was confirmed in the DomainBed that facilitated the comparison of DG performance in the fair and realistic environment.

2. Related Work

The main goal of domain generalization (DG) is to generate domain-invariant features so that the model is generalizable to unseen target domains, which are generally outside the training distribution. Of a landmark work, Vapnik *et al.* [40] introduces Empirical Risk Minimization (ERM) that minimizes the sum of errors across domains. Notable variants have been introduced to learn domain-invariant features by matching distributions across different domains. Ganin *et al.* [15] utilizes an adversarial network to match such distributions, while Li *et al.* [29] instead matches the conditional distributions across domains. Such a shared feature space is optimized by minimizing maximum mean discrepancy [28], transformed feature distribution distance [31], or covariances [38]. In this work, we also follow this stream of work, but we explore the benefit of self-supervised contrastive learning that can inherently learn to domain-invariant discriminating feature by explicitly mapping the "same-class" latent representations close together.

To our best knowledge, there are few that applied contrastive learning in the domain generalization setting. Classification and contrastive semantic alignment (CCSA) [30] and model-agnostic learning of semantic features (MASF) [11] aimed to reduce the distance of same-class (positive pair) feature distributions while increasing the distance of different-class (negative pair) feature distributions. However, using such negative pairs often make the training unstable unless useful negative samples are available in the same batch, which is often challenging. To address this issue, we focus on minimizing a distance between the same-class (positive pair) features in the embedding space as recently studied for the self-supervised learning task [7, 20, 33], including BYOL [17] and SimSiam [8].

Inter-domain mixup [45, 44, 43] techniques are introduced to perform empirical risk minimization on linearly interpolated examples from random pairs across domains. We also utilize such a mixup, but we only interpolate same-class features to preserve the class-specific features. We observe that such a same-class mixup help obtaining robust performance for unseen domain data.

As another branch, JiGen [5] utilizes a self-supervised signal by solving a jigsaw puzzle as a secondary task to improve generalization. Meta-learning frameworks [27] are also explored for domain generalization to meta-learn how to generalize across domains by leveraging MAML [14]. Some also explored splitting the model into domain-invariant and domain-variant components by low-rank parameterization [26], style-agnostic network [32], and domain-specific aggregation modules [12].

3. Method

We start by motivating our method before explaining its details. The main goal of domain generalization is to learn a domain-invariant representation from multiple source domains so that a model can generalize well across unseen

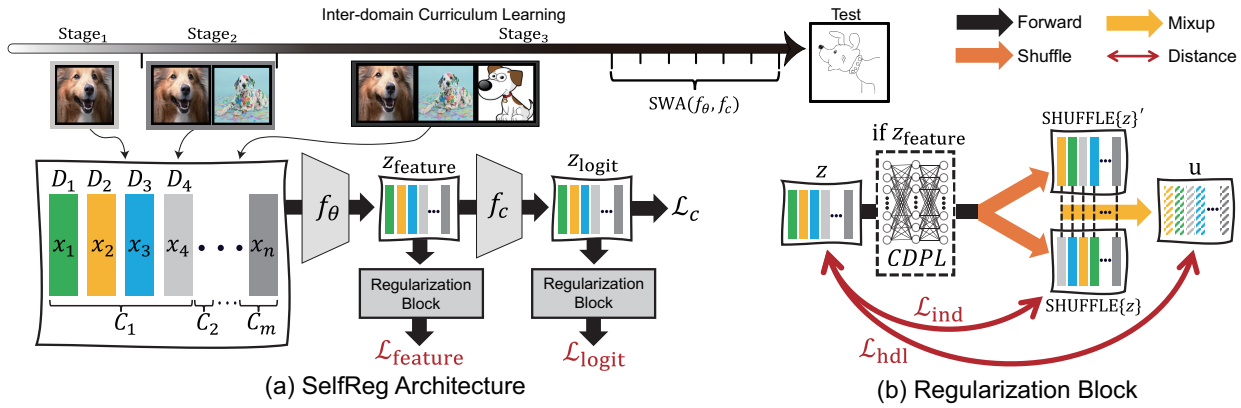

Figure 2. An overview of our proposed SelfReg. We propose the self-supervised (in-batch) contrastive losses to regularize the model to learn domain-invariant representations. These losses regularize the model to map the representations of the "same-class" samples close together in the embedding space. We compute the following two dissimilarities in the embedding space: (i) individualized and (ii) heterogenerous self-supervised dissimilarity losses. We further use the stochastic weight average (SWA) technique and the inter-domain curriculum learning (IDCL) to optimize gradients in conflict directions.

target domains. While domain-variant representation can be achieved to some degree through deep network architectures, invariant representations are often harder to achieve and are usually implicitly learned with the task. To address this, we argue that a model should learn a domain-invariant discriminating feature by comparing among different samples – the comparison can be performed between positive pairs of same-class inputs and negative pairs of different-class inputs.

Here we propose the self-supervised contrastive losses to regularize the model to learn domain-invariant representation by mapping the representations of the "same-class" samples close together, while that of "different-class" samples further away in the embedding space. This may share a similar idea with contrastive learning, which trains a discriminative model on multiple input pairs according to some notion of similarity. Thus, we start with the recent batch contrastive approaches and extend them to the domain generalization setting. While some domain generalization approaches need to modify the model architecture during learning, our proposed contrastive method is much simpler where no modification to the model architecture is needed.

In the next section, we explain our proposed self-supervised contrastive losses for domain generalization tasks, which mainly measures the following two feature-level dissimilarities in the embedding space: (i) Individualized In-batch Dissimilarity Loss (Section 3.1) and (ii) Heterogeneous In-batch Dissimilarity Loss (Section 3.2). Note that these losses can be applied to both the intermediate features and the logits from the classifier (Section 3.3). In fact, in our ablation study (Section 4.4), the combined use of both regularization achieves the best performance. In Section 3.4, we also discuss the stochastic weight average (SWA) technique that we use with our self-supervised contrastive losses and observe a further performance improvement, which is possibly due to SWA provides the more flatness in loss surface by ensembling domain-specific models. Lastly, in Section 3.5, we discuss the inter-domain curriculum learning (IDCL) strategy so that examples from source domains are exposed in a meaningful order to gradually provide more complex ones.

3.1. Individualized In-batch Dissimilarity Loss

Given latent representations $\mathbf{z}_i^c = f_\theta(\mathbf{x}_i)$ for a class label $c \in \mathcal{C}$ and $i \in \{1, 2, \ldots, N\}$, we compute the individualized in-batch dissimilarity loss $\mathcal{L}_{\text{ind}}$. Note that we use a feature generator f_θ parameterized by θ and we use a batch size of N. The dissimilarity between a positive pair of the "same-class" latent representations is measured as in the following Eq. 1:

$$\mathcal{L}_{\text{ind}}(\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N} \left|\left|\mathbf{z}_i^c - f_{\text{CDPL}}\left(\mathbf{z}_{j\in[1,N]}^c\right)\right|\right|_2^2 \qquad (1)$$

where $\mathbf{z}_j^c$ is randomly chosen from other in-batch latent representations $\{\mathbf{z}_i^c\}$ that has the same class label $c \in \mathcal{C}$. Note that we only consider optimizing the alignment of positive pairs and the uniformity of the representation distribution at the same time. As discussed in [17], we use an additional MLP layer f_{CDPL}, called Class-specific Domain Perturbation Layer, to prevent the performance drop caused by so-called representation collapse. We provide an ablation study in Section 4.4 to confirm the use of f_{CDPL} achieves better performance.

For better computational efficiency, we use the following two steps to find all positive pairs. (i) We first cluster and

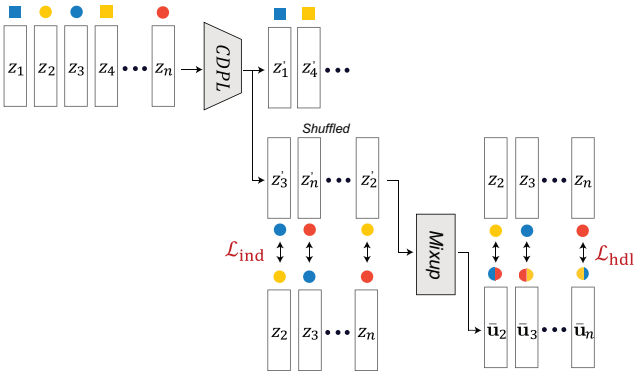

Figure 3. An overview of our proposed self-supervised contrastive regularization losses.

order latent representations $\mathbf{z}_i$ into a same-class group, i.e. $\{\mathbf{z}_i^c\}$ for $c \in \mathcal{C}$. (ii) For each same-class group, we modify its order by random shuffling and obtain $\texttt{SHUFFLE}\{\mathbf{z}_i^c\}$. (iii) We finally form a positive pair in order from $\{\mathbf{z}_i^c\}$ and $\texttt{SHUFFLE}\{\mathbf{z}_i^c\}$.

3.2. Heterogeneous In-batch Dissimilarity Loss

To further push the model to learn domain-invariant representations, we use an additional loss, called heterogeneous in-batch dissimilarity loss. Given latent representations $\mathbf{u}_i = f_{\text{CDPL}}(\mathbf{z}_i^c)$ from the previous step, we apply a two-domain Mixup layer to obtain the interpolated latent representation $\bar{\mathbf{z}}_i$ across different domains. This regularizes the model on the mixup distribution [46], i.e. a convex combination of samples from different domains. This is similar to a layer proposed by Wang *et al.* [43] as defined as follows:

$$\bar{\mathbf{u}}_i^c = \gamma \mathbf{u}_i^c + (1-\gamma)\mathbf{u}_{j\in[1,N]}^c \tag{2}$$

where $\gamma \sim \texttt{Beta}(\alpha, \beta)$ for $\alpha = \beta \in (0, \infty)$. Similarly, $\mathbf{u}_j^c$ is randomly chosen from $\{\mathbf{u}_i^c\}$ for $i \in \{1, 2, \ldots, N\}$ that have the same class label. Note that $\gamma \in [0, 1]$ is controlled by hyper-parameters α and β.

Finally, we compute the heterogeneous in-batch dissimilarity loss $\mathcal{L}_{\text{hdl}}(\mathbf{z})$ as follows:

$$\mathcal{L}_{\text{hdl}}(\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N} \left|\left|\mathbf{z}_i^c - \bar{\mathbf{u}}_i^c\right|\right|_2^2 \tag{3}$$

3.3. Feature and Logit-level Self-supervised Contrastive Losses

The proposed individualized and heterogeneous in-batch dissimilarity losses can be applied to both the intermediate features and the logits from the classifier. We use the loss function $\mathcal{L}_{\text{SelfReg}}$ as follows:

$$\mathcal{L}_{\text{SelfReg}} = \lambda_{\text{feature}}\mathcal{L}_{\text{feature}} + \lambda_{\text{logit}}\mathcal{L}_{\text{logit}} \tag{4}$$

where we use λ_{feature} and λ_{logit} to control the strength of each term. As we use a linear form of the loss function, which often needs to be properly balanced so that network parameters converge to generate domain-invariant features that are also useful for the original classification task. We observe that our self-supervised contrastive losses $\mathcal{L}_{\text{SelfReg}}$ become dominant after the initial training stage, inducing gradient imbalances to impede proper training. To mitigate this issue, we apply two gradient stabilization techniques: (i) loss clipping and (ii) stochastic weights averaging (SWA), and (iii) inter-domain curriculum learning (IDCL). For (i), we modify gradient magnitudes to be dependent on the magnitude of the classification loss $\mathcal{L}_{\text{c}}$ – i.e. we use the gradient magnitude modifier $\texttt{min}(1.0, \mathcal{L}_{\text{c}})$ and thus $\mathcal{L}_{\text{feature}} = \texttt{min}(1.0, \mathcal{L}_{\text{c}})\left[\gamma\mathcal{L}_{\text{ind}} + (1-\gamma)\mathcal{L}_{\text{hdl}}\right]$. This technique is effective to dynamically balance these losses during training. For (ii) and (iii), we discuss details respectively in Section 3.4 and in Section 3.5.

Loss Function Ultimately, we use the following loss function $\mathcal{L}$ that consists of classification loss $\mathcal{L}_{\text{c}}$ as well as our self-supervised contrastive loss $\mathcal{L}_{\text{SelfReg}}$:

$$\mathcal{L} = \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{SelfReg}} \tag{5}$$

3.4. Stochastic Weights Averaging (SWA)

Stochastic Weight Average (SWA) is an ensembling technique to find a flatter minimum in loss space by averaging snapshots of model parameters derived from multiple local minima in the training procedure [23]. It is known that finding a flatter minima guarantees better generalization performance [19], and thus it has been used in domain adaptation and generalization fields that require high generalization performance [48, 6].

Given model weight space $\Omega = \{\omega_0, \omega_1, \ldots, \omega_N\}$, where N is the number of training steps. There is no specific constraint for sampling model weights, however, in general, sampling process is performed at a specific period while the model is sufficiently converged. We use c as a cyclic step length and sample weight space for SWA is $\Omega_{\text{swa}} = \{\omega_{m+kc}\}$ for $k \geq 0, 0 \leq m \leq m+kc \leq N$, where m indicates the initial step for SWA. Then we can derive the averaged weight w_{swa} as follows:

$$\omega_{\text{swa}} = \frac{1}{k+1}\sum_{i=0}^{k}\omega_{m+ic}. \tag{6}$$

3.5. Inter-domain Curriculum Learning (IDCL)

Leveraging the ImageNet-pretrained ConvNet as a backbone is a common practice in the domain generalization literature. Such models are then often fine-tuned with examples that are randomly presented from all source domains $\mathcal{D}_i$ for $i \in \{1, 2, \ldots, M\}$, which often make the training unstable as it optimizes gradients in conflict directions.

Here we use a curriculum learning strategy where source domains are exposed in a meaningful order to gradually

provide more complex ones during training. We first arrange M source domains in a sequence ordered by a distance from a domain where a backbone is pre-trained, e.g. ImageNet [9] data set. Given the ordered source domains $\mathcal{D}'_i$ for $i \in 1, 2, \ldots, M$, we divide the overall training process into M sub-stages $s \in \{1, 2, \ldots, M\}$. At each stage s, a model is only exposed to a subset of source domains, i.e. $\{\mathcal{D}'_{i \leq s}\}$ – a model is exposed to gradually learn more complex examples.

We observe such a curriculum learning-based training strategy provides the substantial performance improvement possibly due to regularizing conflict gradients. We provide details in Section 4.4.

4. Experiments

4.1. Implementation and Evaluation Details

Following Huang *et al.* [22], we train our model, for approximately 30 epochs, with a SGD optimizer using ResNet18 [21] as a backbone, which is pretrained on ImageNet [9]. Our backbone produces 512-dimensional latent representation from the last layer. The batch size is set to 128 and learning rate to 0.004, which is decayed to 0.1 at 24 epochs. Note that such a decaying learning rate is not used when it combined with the Stochastic Weights Averaging technique, where we instead compute the averaged weight w_{swa} at the every end of each epoch. The loss weights are $\lambda_{\text{feature}} = 0.3$ and $\lambda_{\text{logit}} = 1.0$ were determined using grid-search. We provide our experimental results of our hyper-parameters tuning in the supplemental material. For a two-domain Mixup layer, we use $\alpha = \beta = 0.5$. The model architecture for the class-specific domain perturbation layer f_{CDPL} is a 2-layer MLPs with the number of hidden units set to 1024, where we apply batch normalization followed by ReLU activation function. Following RSC [22], data augmentation is used in our experiments to improve model generalizability. This is done by randomly cropping, flipping horizontally, jittering color, and changing the intensity.

Dataset To verify the effectiveness of the proposed method, we evaluate our proposed method on the publicly available PACS [26]. This benchmark dataset contains the overall 10k images from four different domains: $Photo$, $Art\ Painting$, $Cartoon$, and $Sketch$. This dataset is particularly useful in domain generalization research as it provides a bigger domain shift than existing photo-only benchmarks. This dataset provides seven object categories: i.e. dog, elephant, giraffe, guitar, horse, house, and person. We follow the same train-test split strategy from [26], we split examples from training domains to 9:1 (train:val) and test on the whole held-out domain. Note that we use the best-performed model on validation for testing.

Table 1. Image recognition accuracy (%) comparison with the state-of-the-art approach, RSC [22], on PACS [26] test set. We also report standard deviation from a set of 20 models individually trained for each model and each test domain.

Model	Test Domain				Average
	Photo	*Art Painting*	*Cartoon*	*Sketch*	
A. DeepAll	95.66 ± 0.4	79.89 ± 1.3	75.61 ± 1.5	73.33 ± 2.8	81.12 ± 0.8
B. RSC [22]	94.56 ± 0.4	79.88 ± 1.7	76.87 ± 1.2	77.11 ± 2.7	82.10 ± 0.9
C. A + SelfReg (ours)	**96.22 ± 0.3**	**82.34 ± 0.5**	**78.43 ± 0.7**	**77.47 ± 0.8**	**83.62 ± 0.3**

4.2. Performance Evaluation

In Table 1, we first compare our model with the state-of-the-art method, called Representation Self-Challenging (RSC) [22], which iteratively discards the dominant features during training and thus encourages the network to fully use remaining features for the final verdict. For a fair comparison, all models use the identical backbone ConvNet, i.e. ResNet18. We use the leave-one-out setting, i.e. a specific single domain is used as a test domain and the others as a training domain. To see the performance variance, we trained each model 20 times for each test domain and report the average image recognition accuracy and its standard deviation. As shown in Table 1, our proposed model clearly outperforms the other approaches in all test domains (compare the model B vs. model C), and the average image recognition accuracy is 1.52% better than RSC [28], while produces lower model variance (0.9 vs. 0.3 on average).

Qualitative Analysis by t-SNE We use t-SNE [39] to compute pairwise similarities in the latent space and visualize in a low dimensional space by matching the distributions by KL divergence. In Figure 4, we provide a comparison of t-SNE visualizations of baseline, RSC, and ours. The better a model generalizes well, the points in the t-SNE should be more clustered. As shown in Figure 4, (a) the baseline model and (b) RSC [22] produce scattered multiple clusters for each domain and class (see houses in the different clusters according to their domain). Ours is not the case for this. As shown in Figure 4 (c), objects from the same class tend to form a merged cluster, making latent representations close to each other in the high-dimensional space.

The Effect of Dissimilarity Loss We propose two types of self-supervised contrastive loss that map the "same-class" samples close together. We observe in Figure 5 that "same-class" pairwise distance is effectively regularized in both latent (a) feature and (b) logit space (compare dotted (baseline) vs. red solid line (ours)). This was not the case for the baseline. Note that we use Euclidean-based distance to measure the pairwise difference.

Analysis with GradCAM We use GradCAM [37] to visualize image regions where the network attends to. In Fig. 6, we provide examples for different target domains where we

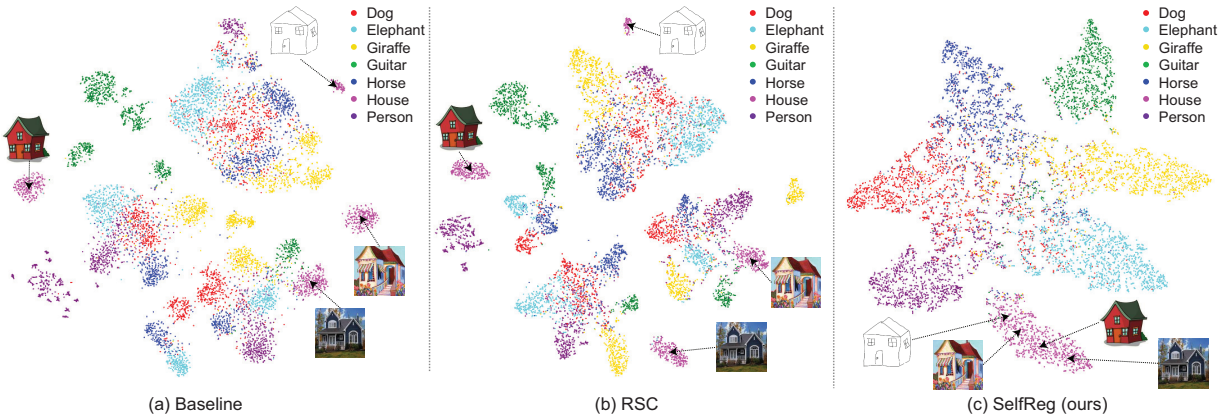

Figure 4. Visualizations by t-SNE [39] for (a) baseline (no DG techniques), (b) RSC [22], and (c) ours. We extract latent representations from each model in leave-one-out setting, and then visualize them. For better understanding, we also provide sample images of house from all target domains. Note that we differently color-coded each points according to its class. *Data*: PACS [26]

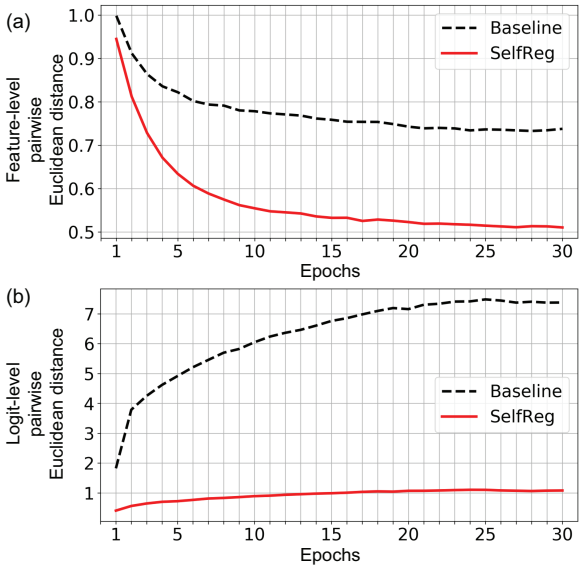

Figure 5. Distance between (a) a pair of same-class features and (b) a pair of same-class logits. We measure such distance at every epoch during training and compare ours (solid red) with baseline (dotted). Euclidean-based distance is used to measure distance in feature space. *Data*: PACS [26]

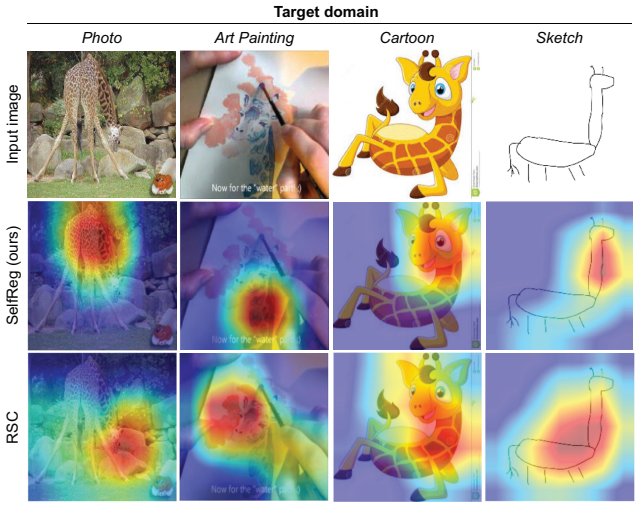

Figure 6. Original images with a giraffe for different domains (1st row). We provide visualizations of Grad-CAM [37] for ours and RSC [22] , which localizes class-discriminative regions. *Data*: PACS [26]

compare the model's attention maps. We observe ours better captures the class-invariant feature (i.e. the long neck of the giraffe), while RSC [22] does not. Red is the attended region for the network's final verdict. We will provide more diverse examples in the supplemental material.

4.3. Single-source Domain Generalization

We also evaluate our model in an extreme case for the domain generalization task. We train our model with examples from a single source domain (not multiple source domains as we see in a previous experimental setting), and then we evaluate with examples from other remaining target domains. As shown in Table 2, we report scores for all source-target combinations, i.e. rows and columns for source and target domains, respectively. As a baseline, we compare ours with those of RSC [22] evaluated in the same setting (compare scores in left and right tables). We also report their differences in the last row ('+' indicates that ours performs better). We observe in Table 2 that ours generally outperform alternative, where the average accuracy is improved by 0.93%.

4.4. Ablation Study

In Table 3, we compare variants of our model by removing each component: i.e. (i) feature-level in-batch dissim-

Table 2. As an extreme case for the domain generalization task, we train our model with a single source domain (rows) and evaluate with other remaining target domains (columns). As a baseline, we also compare with RSC [22] of the same setting (compare left and right tables). We also report their differences in the last row (+ indicates that ours performs better).

RSC [22]	Target domain					**SelfReg**	Target domain				
	Photo	*Art Painting*	*Cartoon*	*Sketch*	Average		*Photo*	*Art Painting*	*Cartoon*	*Sketch*	Average
Photo	-	66.33 ± 1.8	26.47± 2.5	32.08 ± 2.0	41.63 ± 1.6	*Photo*	-	67.72 ± 0.7	28.97 ± 1.0	33.71 ± 2.6	43.46 ± 1.1
Art Painting	96.28 ± 0.4	-	62.54 ± 2.1	53.19 ± 3.2	70.67 ± 1.2	*Art Painting*	96.62 ± 0.3	-	65.22 ± 0.7	55.94 ± 3.1	72.59 ± 1.1
Cartoon	85.89 ± 1.1	68.99 ± 1.4	-	70.38 ± 1.7	75.08 ± 1.0	*Cartoon*	87.53 ± 0.8	72.09 ± 1.2	-	70.06 ± 1.6	76.56 ± 0.8
Sketch	47.40 ± 3.5	37.99 ± 1.4	56.36 ± 3.0	-	47.25 ± 2.9	*Sketch*	46.07 ± 5.3	37.17 ± 4.0	54.03 ± 3.2	-	45.76 ± 3.8
Average	76.52	57.77	48.45	51.88	58.66	Average	**76.74 (+0.22%)**	**58.99 (+1.22%)**	**49.41 (+0.96%)**	**53.24 (+1.36%)**	**59.59 (+0.93%)**

Table 3. Ablation study of SelfReg on PACS. *Abbr.* R_f: feature-level in-batch dissimilarity loss, R_l: logit-level in-batch dissimilarity loss, Mix-up: two-domain mix-up layer, CDPL: class-specific domain perturbation layer, SWA: stochastic weights averaging, IDCL: inter-domain curriculum learning

Model	Components						Test Domain				Average
	$\mathcal{L}_{\text{logit}}$	$\mathcal{L}_{\text{feature}}$	Mixup	CDPL	SWA	IDCL	*Photo*	*Art Painting*	*Cartoon*	*Sketch*	
A. SelfReg (ours)	✓	✓	✓	✓	✓	✓	**96.22 ± 0.3**	**82.34 ± 0.5**	**78.43 ± 0.7**	**77.47 ± 0.8**	**83.62 ± 0.3**
B. A w/o IDCL	✓	✓	✓	✓	✓		96.09 ± 0.3	81.89 ± 0.6	78.03 ± 0.4	77.21 ± 1.1	83.30 ± 0.3
C. B w/o SWA	✓	✓	✓	✓			96.10 ± 0.5	81.43 ± 1.0	77.86 ± 1.0	76.81 ± 1.2	83.05 ± 0.5
D. C w/o CDPL	✓	✓	✓				96.04 ± 0.4	81.66 ± 1.3	77.48 ± 1.2	76.16 ± 1.3	82.84 ± 0.6
E. D w/o Mixup	✓	✓					96.05 ± 0.3	81.77 ± 1.1	77.45 ± 1.1	75.74 ± 1.6	82.75 ± 0.7
F. E w/o $\mathcal{L}_{\text{feature}}$	✓						96.19 ± 0.3	81.59 ± 1.2	76.98 ± 1.3	75.71 ± 1.3	82.62 ± 0.5
G. F w/o $\mathcal{L}_{\text{logit}}$ (baseline)							95.66 ± 0.4	79.89 ± 1.3	75.61 ± 1.5	73.33 ± 2.8	81.12 ± 0.8

ilarity regularization, (ii) logit-level in-batch dissimilarity regularization, (iii) a two-domain mixup layer, (iv) a class-specific domain perturbation layer (CDPL), (v) stochastic weights averaging (SWA), and (vi) inter-domain curriculum learning (IDCL).

Effect of Inter-domain Curriculum Learning (IDCL) We observe in Table 3 that applying our inter-domain curriculum learning (IDCL) provides the recognition accuracy (compare model A vs. B). Scores are generally improved in all target domains, i.e. the average accuracy is improved by 0.32%. With IDCL, source domains are ordered by a distance from a domain where a backbone is pre-trained, and specifically, we arrange source domains in a sequence ordered by image recognition accuracy on the validation set in the leave-oneout setup – which uses a single domain as a test domain and the others as a training domain. Thus, we only provide examples from $\mathcal{D}_{\text{1st stage}} \in \{\textit{Photo}\}$ for the first 5 epochs, which has a weak domain-shift from our ImageNet pre-trained model. We then provide examples from $\mathcal{D}_{\text{2nd stage}} \in \{\textit{Photo}, \textit{Art Painting}\}$, and then from all source domains for the remaining epochs.

Effect of Stochastic Weights Averaging (SWA) As shown in Table 3, the use of stochastic weight average technique further provides better performance (compare model B vs. C) in all target domains, i.e. the average accuracy is improved by 0.25%. This is probably due to SWA provides the flatness in loss surface by ensembling domain-specific models, which generally have multiple local-minima during the training procedure.

Effect of Mixup and CDPL As shown in Table 3, we observe that both CDPL and Mixup components contribute to improve the overall performance (compare Model C vs. D for CDPL, and Model D vs. E for Mixup). Such improvement is more noticeable for the *Sketch* domain, which may support that CDPL reinforces the overall effect of mixup and makes DG performance more robust for target domains that are significantly distanced from their source domains.

Feature- and Logit-level Contrastive Losses Model F, as defined as the baseline model (Model G) plus $\mathcal{L}_{\text{logit}}$, had an average performance improvement of 1.50%. Accuracy improved and variance decreased across all of the domains. Therefore, regularization to minimize the logit vector-wise distance on positive pairs appears effective in extracting domain invariant features. Furthermore, Model E, which adds $\mathcal{L}_{\text{feature}}$ and $\mathcal{L}_{\text{logit}}$ to the baseline model, exhibited even greater performance increase. Minimizing feature distances of positive pairs as well as logit distances, was observed to be effective in improving DG performance.

Table 4. Average out-of-distribution test accuracies on the DomainBed setting. Here we compare 14 domain generalization algorithms in the exact same conditions. Note that we train domain validation set as a model selection method. †: Ours does not use IDCL technique due to implementational inflexibility on the DomainBed environment. *Abbr.* D: learning domain-invariant features by matching distributions across different domains, A: adversarial learning strategy, M: inter-domain mix-up, C: contrastive learning, U: unsupervised domain adaptation, which is originally designed to take examples from the target domain during training.

Model	D	A	M	C	U	CMNIST [1]	RMNIST [16]	VLCS [13]	PACS [26]	OfficeHome [42]	TerraIncognita [2]	DomainNet [34]	Average
SelfReg† (ours)	✓		✓	✓		51.6 ± 0.2	98.0 ± 0.1	77.5 ± 0.0	86.5 ± 0.3	69.4 ± 0.2	51.0 ± 0.4	44.6 ± 0.1	68.4
SelfReg† (no SWA)	✓		✓	✓		52.1 ± 0.2	98.0 ± 0.1	77.8 ± 0.9	85.6 ± 0.4	67.9 ± 0.7	47.0 ± 0.3	41.5 ± 0.2	67.1
CORAL [38]	✓				✓	51.5 ± 0.1	98.0 ± 0.1	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	67.5
SagNet [32]	✓	✓				51.7 ± 0.0	98.0 ± 0.0	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	48.6 ± 1.0	40.3 ± 0.1	67.2
Mixup [45]			✓			52.1 ± 0.2	98.0 ± 0.1	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	66.7
MLDG [27]						51.5 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	66.7
ERM [41]						51.5 ± 0.1	98.0 ± 0.0	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	66.6
MTL [3]						51.4 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	45.6 ± 1.2	40.6 ± 0.1	66.2
RSC [22]						51.7 ± 0.2	97.6 ± 0.1	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	46.6 ± 1.0	38.9 ± 0.5	66.1
ARM [47]						56.2 ± 0.2	98.2 ± 0.1	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	66.1
DANN [15]	✓	✓			✓	51.5 ± 0.3	97.8 ± 0.1	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	66.1
VREx [25]	✓					51.8 ± 0.1	97.9 ± 0.1	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	65.6
CDANN [29]	✓	✓				51.7 ± 0.1	97.9 ± 0.1	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3	65.6
IRM [1]						52.0 ± 0.1	97.7 ± 0.1	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	65.4
GroupDRO [36]	✓					52.1 ± 0.0	98.0 ± 0.0	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	64.8
MMD [28]	✓					51.5 ± 0.2	97.9 ± 0.0	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	63.3

5. Large-scale Experiments on DomainBed

We further conduct experiments using DomainBed [18], which is a unified testbed useful for evaluating domain generalization algorithms. This testbed currently provides seven multi-domain datasets (i.e. ColoredMNIST [1], RotatedMNIST [16], VLCS [13], PACS [26], OfficeHome [42], TerraIncognita [2], and DomainNet [34]) and provides benchmarks results of 14 baseline approaches (i.e. ERM [41], IRM [1], GroupDRO [36], Mixup [45], MLDG [27], CORAL [38], MMD [28], DANN [15], CDANN [29], MTL [3], SagNet [32], ARM [47], VREx [25], and RSC [22]).

As shown in Table 4, we also report scores for our model evaluated in the setting of DomainBed. We observe in Table 4 that ours generally shows matched or better performance against alternative state-of-the-art methods, where ours is ranked top places in terms of average of all seven benchmarks. Note that ours does not use IDCL (see Section 3.5) technique, which we believe that further improvements are highly achievable combined with this technique. We provide more detailed scores for each domain in the supplemental material. Note that DANN [15] and CORAL [38] are designed to take examples from the target domain during training – i.e. CORAL [38] is trained to minimize the distance between covariances of the source and target features. Note also that some studies [32, 15, 29] use the adversarial learning setting to obtain an unknown domain-invariant feature by fitting implicit generative models, such as GAN (generative adversarial networks). Though GAN is a powerful framework, the alternating gradient updates procedure is often highly unstable and often results in mode collapse [24].

6. Conclusion

In this paper, we proposed SelfReg, a new regularization method for domain generalization that leverages a self-supervised contrastive regularization loss with only positive data pairs, mitigating problems caused by negative pair sampling. Our experiments on PACS dataset and DomainBed benchmarks show that our model matches or outperforms prior work under the standard domain generalization evaluation setting. In future work, it would be interesting to extend SelfReg with the siamese network, enabling the model to choose better positive data pairs.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2021-0-00994, Sustainable and robust autonomous driving AI education/development integrated platform). J. Kim is partially supported by the National Research Foundation of Korea grant (NRF-2021R1C1C1009608), Basic Science Research Program (NRF-2021R1A6A1A13044830), and ICT Creative Consilience program (IITP-2021-2020-0-01819).

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv preprint arXiv:1907.02893, 2019. 8

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In Proceedings of the European Conference on Computer Vision (ECCV), pages 456–473, 2018. 8

[3] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. arXiv preprint arXiv:1711.07910, 2017. 8

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. Advances in neural information processing systems, 24:2178–2186, 2011. 1

[5] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2229–2238, 2019. 2

[6] Junbum Cha, Hancheol Cho, Kyungjae Lee, Seunghyun Park, Yunsung Lee, and Sungrae Park. Domain generalization needs stochastic weight averaging for robustness on domain shifts, 2021. 4

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning (ICML), pages 1597–1607. PMLR, 2020. 1, 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566, 2020. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255. Ieee, 2009. 5

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. A deep convolutional activation feature for generic visual recognition. UC Berkeley & ICSI, Berkeley, CA, USA. 1

[11] Qi Dou, Daniel C Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. arXiv preprint arXiv:1910.13580, 2019. 2

[12] Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In German Conference on Pattern Recognition, pages 187–198. Springer, 2018. 2

[13] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1657–1664, 2013. 8

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

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. The journal of machine learning research, 17(1):2096–2030, 2016. 1, 2, 8

[16] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2551–2559, 2015. 8

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733, 2020. 2, 3

[18] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. arXiv preprint arXiv:2007.01434, 2020. 2, 8

[19] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. 4

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9729–9738, 2020. 1, 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. 5

[22] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In Proceedings of the European Conference on Computer Vision (ECCV), 2020. 5, 6, 7, 8

[23] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407, 2018. 4

[24] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. arXiv preprint arXiv:1705.07215, 2017. 8

[25] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). arXiv preprint arXiv:2003.00688, 2020. 8

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Deeper, broader and artier domain generalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 2, 5, 6, 8

[27] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for

domain generalization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018. 2, 8

[28] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5400–5409, 2018. 1, 2, 5, 8

[29] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 624–639, 2018. 1, 2, 8

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

[31] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Proceedings of the International Conference on Machine Learning (ICML), pages 10–18. PMLR, 2013. 1, 2

[32] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2, 8

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 1, 2

[34] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1406–1415, 2019. 8

[35] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1278–1286, 2015. 1

[36] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731, 2019. 8

[37] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 618–626, 2017. 5, 6

[38] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 443–450. Springer, 2016. 1, 2, 8

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008. 5, 6

[40] V Vapnik. Statistical learning theory new york. NY: Wiley, 1998. 1, 2

[41] Vladimir N Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5):988–999, 1999. 8

[42] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5018–5027, 2017. 8

[43] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3622–3626. IEEE, 2020. 2, 4

[44] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6502–6509, 2020. 2

[45] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. arXiv preprint arXiv:2001.00677, 2020. 2, 8

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017. 4

[47] Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. arXiv preprint arXiv:2007.02931, 2020. 8

[48] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In Proceedings of the International Conference on Machine Learning (ICML), pages 7523–7532. PMLR, 2019. 4