SimDE: A Simple Domain Expansion Approach for Single-source Domain Generalization

Anonymous L3D-IVU submission

Paper ID 11

Abstract

Single domain generalization challenges model generalizability to unseen target domains when only one source domain is provided for training. To tackle this problem, domain expansion is adopted to learn domain-invariant information by exposing the model to more domain variations, which is still under-explored in previous work. In this paper, we propose a new and simplified objective for learning the desirable domain expansions by generating unconfident samples through the combination of entropy maximization and cross-entropy minimization. We further devise a novel framework that trains a pair of generators from different views by switching the guidance from the dual classifiers. In this way, the resulting method called Simple Domain Expansion (SimDE) can learn diverse domain expansions effectively and efficiently. Extensive experiments on prevalent single domain generalization benchmarks demonstrate the superiority of our method by offering improved results over the state-of-the-arts methods.

1. Introduction

Deep learning models are known to suffer from severe performance degradation in the presence of domain shift [2, 52], where training and test data come from different distributions. Domain generalization is thus proposed, to make the models generalizable to unknown target domains [22, 26, 53]. A classical domain generalization setting assumes the access to multiple training source domains at training, *i.e.,* multi-source domain generalization (MultiDG). Over the last decades, significant progress has been made for MultiDG, resorting to techniques such as domain alignment [28, 30, 34, 36], meta learning [1, 11, 25, 31], domain augmentation [33, 42, 53, 58, 59], and self-supervisions [3, 4, 50].

However, collecting and labelling data from multiple domains can be costly, which makes MultiDG less practical in real world. A more challenging and realistic, but less studied setting, is single-source domain generalization (SingleDG), where only one source domain is available at training. A single source domain impedes the model from learning domain-invariant information, as no domain comparison may be made. Consequently, the model can easily overfit the domain-specific signals on the single source domain.

The most prevalent way for tackling SingleDG is domain expansion [29, 40, 41, 47, 51, 57], which attempts to expand the single source domain by generating pseudo domains. The key of domain expansion is how to diversify the domain distributions under the strong constraint of preserving original semantics. Previous methods have made large progresses in learning diverse domain distributions, but two limitations are still existed. Firstly, to balance the domain diversification and semantic preserving behavior, the main objective for domain expansion usually involves a complicated interaction of different loss terms defined at the input, latent and output spaces respectively, and need to be tuned carefully. Secondly, recent work train many generators to learn domain expansion from different views, *e.g.*, PDEN stacks 20 generators progressively [29], while L2D uses 5 generators in parallel with different kernel sizes [51].

To deal with the above two limitations, we propose a simple yet more effective way for domain expansion in this paper. We first seek to simplify the overly complicated domain expansion learning objective in previous work. Inspired by the positive correlation between the output entropy of the classifier and the extent of domain shift [32, 48], we expand the source domain distribution by maximizing the predicted entropy of the generated samples. In other words, the desired domain expansions can be obtained by generating the unconfident samples with respect to the classifier. Meanwhile, we constrain the semantics by minimizing the cross-entropy loss between the generated samples and the original labels. In this way, the main objective for domain expansion can be implemented as a simple trade-off between the entropy loss maximization and cross-entropy loss minimization. Moreover, both the entropy and cross-entropy can be easily calculated at the logit level of the classifier, without the requirement of latent feature manipu-

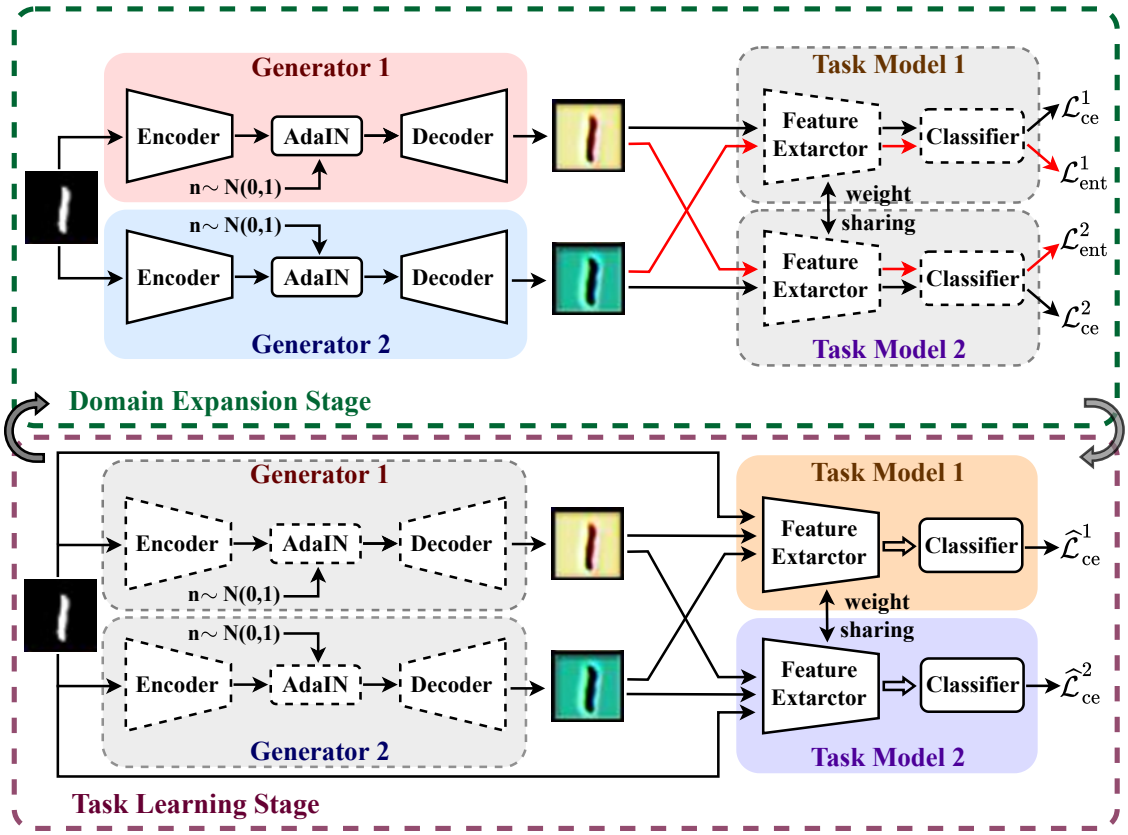


Figure 1. The illustration of our method SimDE. The whole learning process iterates between the domain expansion stage and the task learning stage. Parameters of the dashed model blocks are fixed in the corresponding stages. Forward propagation that related to loss maximization is highlighted in red.

lation or specific model structures [40, 51].

Based on the new objective, we further propose a novel and simple framework that utilizes the classifier discrepancy to train a pair of generators. Specifically, we adopt dual classifiers to construct two mirrored objectives to train the two generators from different views. For training the first generator, the cross-entropy loss of the first classifier is minimized for semantic preservation and the entropy loss of the second classifier is maximized for domain diversification. In contrast, for training the second generator, the classifiers for entropy maximization and cross-entropy minimization are switched. Intuitively, for one specific generator, one of the classifiers acts as its adversary that should be fooled, and the other classifier acts as the referee to prevent the generator taking the shortcuts (*e.g.*, generate noisy samples), while for another generator, the roles of adversary and referee are switched. With these two mirrored objectives, we can obtain sufficiently diverse domain expansions with only two generators without introducing more of them.

After the generators are trained, both the generated and original samples are then used for training the classifiers. We also incorporate two auxiliary components in our framework for further improvement. One is a style divergence loss [29] to prevent each generator from producing samples with collapsed styles. The other is a domain alignment loss to enhance the feature invariance between the original and generated domains. The full method is named as *simple domain expansion* (SimDE) thereafter. The whole training process of SimDE alternates between the domain expansion stage that learns the generators, and the task learning stage that learns the task models. With our method, the final task model can be more generalizable to the unseen domains.

We summarize our main contributions as follows:

- We propose a simple and new objective for domain expansion by maximizing the predicted entropy for domain diversification and minimizing the cross-entropy for semantic preservation.
- We further propose a simple and novel framework which trains only two generators for multi-view domain expansion under the guidance of dual classifiers.
- We evaluate our method on several prevalent SingleDG benchmarks and the results show that our method can reach the state-of-the-arts results.

2. Related Work

Multi-source domain generalization (MultiDG): Massive efforts have been made for MultiDG over the past decades. Previous works mainly resort to domain alignment [28, 30, 34, 36], meta learning [1, 11, 25, 31], domain augmentation [33, 42, 53, 58, 59] or self-supervisions [3, 4, 50]. Representatively, [28, 30, 34] adopt adversarial training [15] to align the feature distributions from different domains, while CCSA [36] uses semantic contrastive losses. MLDG [25] first proposes a meta-learning strategy that simulates MAML [13] by splitting the source domains into disjoint virtual source and virtual target domains. Subsequent work follow this idea to meta learn a regularizer [1], a feature critic loss [31] or the semantic consistency [11]. On the other hand, [33, 42, 58] generate new domains by adversarially confusing the domain classifier to augment the training distributions, while [53, 59] achieve the same goal by interpolating the statistics between domains. Recently, self-supervised tasks, such as the jigsaw puzzle [4, 50] or rotation prediction [3], are incorporated in MultiDG as auxiliary losses to learn generalizable representations. Some other methods also tackle MultiDG through exposing the generic model to the domain-specific counterparts [27], deploying domain-specific masks [5], dedicated style-agnostic regularizations [37] or gradient-based dropout [20].

Single domain generalization (SingleDG): As a more challenging and realistic setting, SingleDG has recently attached attentions from the community. A prevail approach for SingleDG is domain expansion [29, 40, 41, 47, 51, 57], which generates pseudo domains different from the original source domain. Volpi *et al*. [47] first propose to expand the source domain in the manner of adversarial attack [43], and meanwhile restrict the semantics in the feature space. Later, Qiao *et al*. [41] and Zhao *et al*. [57] make improvements by adding additional relaxations through Wasserstein auto-encoders [45] or information bottleneck [44]. Alternatively, Another work of Qiao *et al*. [40] chooses to incorporate uncertainty assessment and MixUp [56]. Recently, Li *et al*. [29] propose a joint framework by progressively expanding the source domains and extracting invariant features through contrastive learning. Wang *et al*. [51] deploy a similar framework that mini-maximizes a mutual information upper bound between source and generated domains. In terms of the methods other than domain expansion, Fan *et al*. [12] adversarially learn adaptive normalization layers, while Cugu *et al*. [8] enforce attention consistency between visual corruptions. Some MultiDG methods [4, 20, 36, 37] without the requirement for multi-source domains can also be applied to SingleDG, but they are usually less effective than the domain expansion-based methods. In this paper, we focus on the perspective of domain expansion, while our method is orthogonal to methods from other perspectives.

3. Method

3.1. Overview

Without loss of generality, we focus on the classification problem of SingleDG in this paper. Given a single source domain $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N}$ with N samples, the goal of SingleDG is to train a domain-agnostic model that can generalize well to the unseen target domains $\mathcal{T}$. To tackle this problem, we follow the main idea of domain expansion by creating samples from pseudo domains to enrich the training domain distributions and enhance domain invariance. We propose a novel and simple method called SimDE to learn the desired domain expansions effectively and efficiently.

The whole framework of SimDE is composed of four networks, namely two task models M_1, M_2 and two generators G_1, G_2, as shown in Figure 1. The task model can be decomposed into a feature extractor and a classification head. The feature extractor is shared across the two task models for complexity reduction. The two generators are responsible for generating samples from pseudo domains in the manner of learned image transformations. The learning process of SimDE alternates between two stages, as shown in Algorithm 1 and Figure 1. In the domain expansion stage, we fix the parameters of the task models and train the generators to learn the desired domain expansions. In the task learning stage, we fix the parameters of the generators and use the generated samples together with the original samples to train the task models. With the two-step training process, the task model can iteratively learn to recognize the hard samples containing different domain variations and thus be more robust to unseen target domains.

3.2. Domain Expansion Stage

We first introduce the learning strategy of SimDE in the domain expansion stage. The goal of domain expansion is to generate novel domains distinct from original domains while not altering the latent semantics. To achieve this goal, we propose a simple objective that operates at the output logit level with a trade-off between predicted entropy maximization and cross-entropy minimization. The expanded domains are encouraged to be distinct from the original domains through entropy maximization, and the latent semantics are constrained to be invariant through cross-entropy minimization. Intuitively, we seek the desired domain expansions by generating the unconfident samples that confuse the current task model.

To learn diverse domain expansions from different views, we further utilize the dual classifier discrepancy to train a pair of generators G_1 and G_2 based on the proposed objective. Specifically, for training G_1, we maximize the entropy loss of the generated samples with respect to the task model M_2 and minimize the corresponding cross-entropy loss with respect to the task model M_1. Conversely,

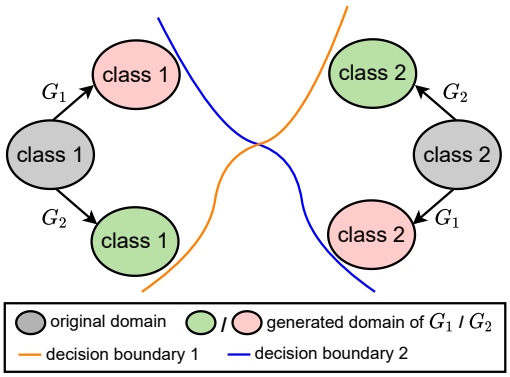


Figure 2. Illustration of multi-view domain expansion in SimDE.

for training generator G_2, we switch the task models responsible for entropy maximization and cross-entropy minimization in the objective. Let the output from different networks be:

$$\begin{aligned} \widehat{x}_i^1 &= G_1(x_i) \quad \widehat{p}_i^{11} = M_1(\widehat{x}_i^1) \quad \widehat{p}_i^{12} = M_2(\widehat{x}_i^1) \\ \widehat{x}_i^2 &= G_2(x_i) \quad \widehat{p}_i^{21} = M_1(\widehat{x}_i^2) \quad \widehat{p}_i^{22} = M_2(\widehat{x}_i^2) \end{aligned} \tag{1}$$

The training objectives are thus formulated as follows:

$$\min_{G_1} -\mathcal{L}_{ent}^2 + \mathcal{L}_{ce}^1 = \frac{1}{N}\sum\nolimits_i (\widehat{p}_i^{12}\log\widehat{p}_i^{12} - y_i\log\widehat{p}_i^{11})$$

$$\min_{G_2} -\mathcal{L}_{ent}^1 + \mathcal{L}_{ce}^2 = \frac{1}{N}\sum\nolimits_i (\widehat{p}_i^{21}\log\widehat{p}_i^{21} - y_i\log\widehat{p}_i^{22})$$

We illustrate the multi-view domain expansion induced by the training objectives in Figure 2. Specifically, generator G_1 is encouraged to expand the domain by approaching the decision boundary of M_2 while maintaining its distance to the decision boundary of M_1. For generator G_2, the domain expansion behavior is the opposite. Due to the discrepancy between the two decision boundaries, the two generators can be guided to independently expand the original source domain from two different directions. Through this way, we can efficiently generate diverse domains with only a pair of generators without introducing more of them.

For each of the generator, we further prevent mode collapse in the generated samples with a style divergence loss as [29]. For generator G_1, the loss is defined as:

$$\mathcal{L}_{sty}^1 = -\frac{1}{N}\sum\nolimits_i ||G_1(x_i, n) - G_1(x_i, n')||_2 \tag{2}$$

where $n, n' \sim N(0,1)$ is Gaussian noises that serve as the conditions for generation, $||\cdot||_2$ is the L2 distance. For generator G_2, the diversity loss $\mathcal{L}_{sty}^2$ is defined similarly.

After incorporating the style divergence loss, the total objectives for domain expansion stage is reformulated as:

$$\min_{G_1} \quad \mathcal{L}_{de}^1 = -\mathcal{L}_{ent}^2 + \mathcal{L}_{ce}^1 + \lambda_{sty}\mathcal{L}_{sty}^1 \tag{3}$$

$$\min_{G_2} \quad \mathcal{L}_{de}^2 = -\mathcal{L}_{ent}^1 + \mathcal{L}_{ce}^2 + \lambda_{sty}\mathcal{L}_{sty}^2 \tag{4}$$

Algorithm 1 The episodic training process of Simple Domain Expansion (SimDE)

Input: Source domain dataset $\mathcal{S}$; task models M_1, M_2; generators G_1, G_2
Output: learned task models M_1, M_2

for all i in $1, \ldots, E_0$ **do**
  initialize the weights of G_1 and G_2 randomly
  for all j in $1, \ldots, E_1$ **do** {domain}
    sample (x_i, y_i) from $\mathcal{S}$
    train G_1 with Eq. 3
    train G_2 with Eq. 4
  end for
  for all k in $1, \ldots, E_2$ **do**
    sample (x_i, y_i) from $\mathcal{S}$
    $(\widehat{x}_i^1, y_i) \leftarrow (G_1(x_i, n), y_i)$
    $(\widehat{x}_i^2, y_i) \leftarrow (G_2(x_i, n), y_i)$
    update M_1 & M_2 with Eq. 8
  end for
end for

where λ_{sty} is the loss balancing weight.

Architecture of the generator: The generator conducts an image-to-image transformation that converts the original image x from the original source domain to a new image $\widehat{x}$ from the unseen pseudo domain. We implement the generator as an encoder-decoder structure with a style integration module based on the adaptive instance normalization (AdaIN) [19], as shown in Figure 1. The scaling parameter β and the shifting parameter γ of AdaIN are derived from a fully connected layer with a random Gaussian noise as the input. The forward process is formulated as:

$$\begin{aligned} &[\beta, \gamma] = \mathrm{FC}(n) \quad \mathrm{AdaIN}(e, n) = \beta \cdot \frac{e - \mu(e)}{\sigma(e)} + \gamma \\ &G(x, n) = \mathrm{Dec}\left(\mathrm{AdaIN}\left(\mathrm{Enc}(x), n\right)\right) \end{aligned} \tag{5}$$

where $n \sim N(0,1)$, μ and σ is the mean and standard deviation, e is the bottleneck embedding, $\mathrm{FC}(\cdot)$, $\mathrm{Enc}(\cdot)$ and $\mathrm{Dec}(\cdot)$ denote the fully-connected layer, the encoder and decoder respectively.

Note that the implementation of the generator is not restricted. Any image-to-image network, such as the spatial transformation network (STN) [21], can be embedded into our framework depending on the tasks at hand. Without loss of generality, we use the AdaIN-based encoder-decoder structure in this paper following [29, 51].

3.3. Task Learning Stage

Our ultimate goal is to obtain a domain-agnostic task model by exposing it to diverse domain variations. Therefore, after training the generators, we freeze their parameters and use them to generate samples from unseen domains.

Each of the task model receives the generated samples from both of the generators, as well as the original samples, and then minimize the cross-entropy loss. Specifically, for task model M_1, the loss is formulated as:

$$\widehat{\mathcal{L}}_{ce}^{1} = -\frac{1}{3N}\sum\nolimits_{i}(y_i \log x_i + y_i \log \widehat{x}_i^1 + y_i \log \widehat{x}_i^2) \quad (6)$$

and the loss $\widehat{\mathcal{L}}_{ce}^{2}$ for task model M_2 can be defined similarly.

To make full use of the generated samples and enhance the representation invariance, we further incorporate a domain-invariant regularization between the original samples and the generated samples, which is also a common practice in previous work [29, 40, 41, 51]. Here we adopt the supervised contrastive loss [23] to increase the mutual information between samples from the same class but different domains. Follow [29, 51], we use a projector head to extract lower dimensional embeddings z for loss computation. Within a specific batch, all the embeddings obtained from different task models and different domains are pooled together as the input of the contrastive loss, which is formulated as follows:

$$\mathcal{L}_{con} = -\sum_{i=0}^{N}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{e^{(z_i\cdot z_p/\tau)}}{\sum_{a\in A(i)}e^{(z_i\cdot z_a/\tau)}} \quad (7)$$

where $A(i)$ if the set of all the samples except z_i, $P(i) = \{p \in A(i) : y_p = y_i\}$ is the set of all positives that share the same class with z_i, and $|P(i)|$ is its carnality, and τ is the temperature parameter.

Taking the cross-entropy loss and the contrastive loss together, we can obtain the total objective during the task learning stage as:

$$\min_{M_1,M_2}\mathcal{L}_{ta} = \widehat{\mathcal{L}}_{ce}^{1} + \widehat{\mathcal{L}}_{ce}^{2} + \lambda_{con}\mathcal{L}_{con} \quad (8)$$

where λ_{con} is the loss balancing weight.

4. Experiment

4.1. Datasets and Setups

To demonstrate the effectiveness of our method, we carry out experiments on four commonly-used SingleDG benchmark datasets: **Digits** is composed by five different datasets, namely MNIST [24], SVHN [38], MNIST-M [14], SYN [14] and USPS [9], with each dataset considered as a unique domain. Follow standard protocols [41, 47], the first 10, 000 images from MNIST are used for training and the remaining four domains are used for testing. **CIFAR10-C** [17] is a corrupted version of CIFAR10 [24]. Follow previous protocols [51, 57], we use the 15 common corruptions for evaluation, which belong to four main categories including weather, blur, noise and digital. **PACS** [26] consists of four domains including photo, art painting, cartoon and sketch. There are totally 9, 991 images and 7 classes. We choose one of the four domains as the source domain and the remaining three domains as the target domains, which results in four different cases. **DomainNet** [39] is a large scale dataset with 6 domains, 345 classes and 596,010 images. Following [8], we use the Real domain as the training set, and the remaining as test.

Table 1. SingleDG accuracy on Digits dataset. All the methods are trained on MNIST. "SV", "MM", "SY", "US" denote SVHN, MNIST-M, SYN and USPS respectively. The bottom half of the table shows the domain expansion-based methods.

Method	SV	MM	SY	US	Avg.
ERM	27.83	52.72	39.65	76.94	49.29
MixUp [56]	28.50	54.00	41.20	76.60	50.10
PAR [49]	30.50	58.40	44.10	76.90	52.50
CCSA [36]	25.89	49.29	37.31	83.72	49.05
d-SNE [54]	26.22	50.98	37.83	93.16	52.05
JiGen [4]	33.80	57.80	43.79	77.15	53.14
AutoAug [6]	45.23	60.53	64.52	80.62	62.72
RandAug [7]	54.77	74.05	59.60	77.33	66.44
ADA [47]	35.51	60.41	45.32	77.26	54.62
M-ADA [41]	42.55	67.94	48.95	78.53	59.49
ME-ADA [57]	42.56	63.27	50.39	81.04	59.32
UMGUD [40]	43.30	67.40	57.10	77.40	61.30
L2D [51]	62.86	**87.30**	63.72	83.97	74.46
PDEN [29]	62.21	82.20	69.39	85.26	74.77
SimDE (ours)	**66.08**	84.90	**70.04**	**86.56**	**76.89**

4.2. Evaluation of Single Domain Generalization

Results on Digits: Table 1 shows the comparison results of our method and state-of-the-arts. The domain expansion-based methods shown in the bottom half generally performs better than other regularization-based methods shown in the top half, justifying the effectiveness of domain expansion for SingleDG. Notably, our method achieves an at least 2.12% average performance gain among the domain expansion competitors. Specifically, the proposed SimDE clearly exceeds the second-best method PDEN with 3.87%, 2.70% and 1.30% on SVHN, MNIST-M and USPS respectively, showing its advantages in overall generalizability.

Results on CIFAR10-C: The average results of the four main corruption categories under severity level-5 are shown in Table 2. Our method reaches the highest average performance among all the competitors and exceeds most of them with large margins. Specifically, the improvement brought by our method is significant on blur and digital corruptions, surpassing the second-best method PDEN with 3.18% and 2.95% respectively. Moreover, PDEN adopts a progressive learning strategy that keeps all the trained generators from previous stages in the memory, while our method is more

Table 2. SingleDG accuracy on CIFAR-10-C. Average results of four main corruption categories at severity level-5 are reported. The bottom half of the table shows the results of the domain expansion-based methods.

Method	Weather	Blur	Noise	Digits	Avg
ERM	67.28	56.73	30.02	62.30	54.08
CCSA [36]	67.66	57.81	28.73	61.96	54.04
d-SNE [54]	67.90	56.59	33.97	61.83	55.07
AutoAug [6]	79.32	74.49	44.90	73.88	69.34
RandAug [7]	**80.36**	78.74	53.78	74.84	72.77
ADA [47]	72.67	67.04	39.97	66.62	61.58
M-ADA [41]	75.54	63.76	54.21	65.10	64.65
ME-ADA [57]	74.44	71.37	66.47	70.83	70.77
L2D [51]	75.98	69.16	73.29	72.02	72.61
PDEN [29]	78.75	75.68	**76.38**	75.54	76.37
SimDE (ours)	79.39	**79.16**	75.30	**78.49**	**78.21**

Table 3. SingleDG accuracy on PACS. For each of the source domain "A", "C", "P" and "S", the averaged results on the remaining target domains are reported. Bottom half of the table shows the results of the domain expansion-based methods.

Methods	A	C	P	S	Avg
ERM	70.49	73.56	41.21	45.92	57.80
JiGen [4]	70.47	73.59	41.05	44.86	57.49
SagNet [37]	73.20	75.67	48.53	50.07	61.87
RSC [20]	74.20	75.27	43.03	51.03	60.88
ADA [47]	72.43	71.97	44.63	45.73	58.69
ME-ADA [57]	74.13	74.53	43.97	54.37	61.75
L2D [51]	77.08	75.21	54.14	55.21	65.41
PDEN [29]	76.43	73.87	58.52	53.92	65.68
SimDE (ours)	**78.52**	**76.14**	**59.32**	**56.39**	**67.59**

efficient by maintaining only two generators.

Results on PACS: We report the comparison results on PACS in Table 3. Methods like SagNet and RSC are proved to be effective under MultiDG scenarios [20, 37], but their performances fall behind the SOTA domain expansion-based methods due to the lack of multiple source domains. Among all the competitors, our method again achieves the highest performance on all the source-target combinations. Specifically, the proposed SimDE surpasses the second-best method PEDN with 2.09%, 2.27%, 0.80%, and 2.47% for source domain art-painting, cartoon, photo and sketch respectively, showing the effectiveness of our method.

Results on DomainNet: We report the comparison results in Table 4. Among all the domain expansion-based competitors, SimDE achieves the best results on 4 out of 5 domains, as well as the top averaged performance. SimDE also obtains better averaged result than the methods based on massive strong augmentations like RandAug, AugMix and VC. Furthermore, SimDE is orthogonal to these strong augmentation methods and can bring further improvements when combined with them. Specifically, when combined with RandAug, SimDE can achieve the highest performance, surpassing the second-best method ACVC with a clear margin of 0.78% on average. These results demonstrate the superiority of SimDE on large-sacle benchmarks.

Table 4. SingleDG accuracy on DomainNet. The models are trained on Real (R), and tested on Painting (P), Infograph (I), Clipart (C), Sketch (S) and Quickdraw (Q). Bottom half of the table shows the results of the domain expansion-based methods.

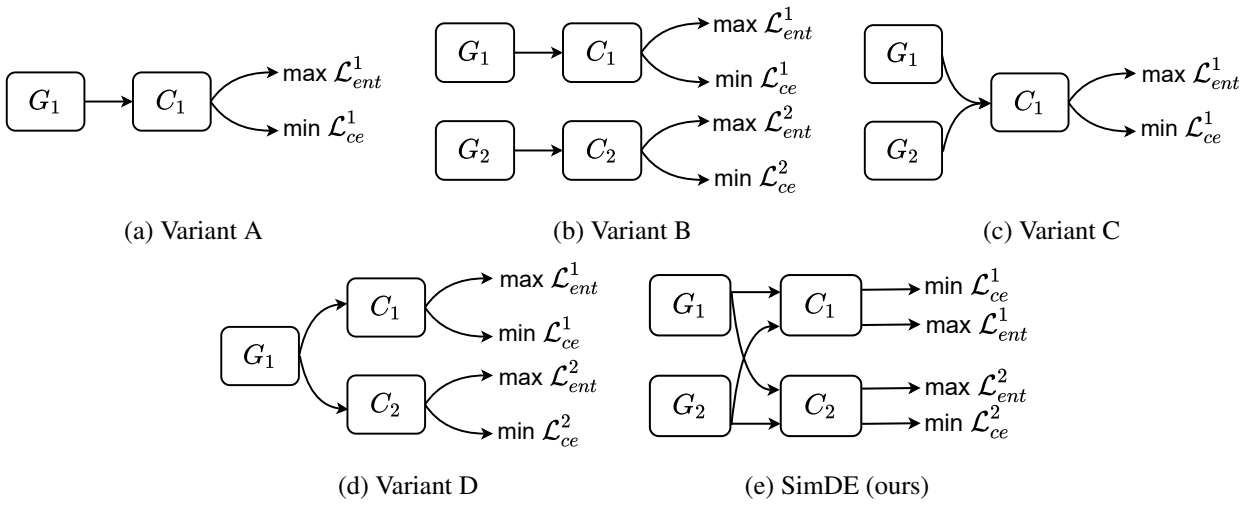
Methods	P	I	C	S	Q	Avg
ERM	38.05	13.31	37.89	26.26	3.36	23.78
MixUp [56]	38.60	**13.94**	38.02	26.01	3.71	24.05
CutOut [10]	38.34	13.69	38.44	26.24	3.65	24.07
CutMix [55]	38.28	13.45	38.65	26.85	3.60	24.17
RandAug [7]	41.30	13.57	41.11	30.40	5.31	26.34
AugMix [18]	40.79	13.89	41.67	29.80	6.26	26.48
VC [8]	41.38	13.58	41.80	30.58	6.06	26.68
ACVC [8]	41.32	12.89	**42.79**	30.86	6.57	26.89
ME-ADA [57]	37.95	13.12	40.31	26.79	4.53	24.54
PDEN [29]	38.45	11.25	38.99	31.71	5.58	25.20
SimDE (ours)	39.96	12.91	41.73	33.46	6.85	26.98
+RA	**41.43**	12.81	42.52	**34.31**	**7.28**	**27.67**
+AugMix	40.73	12.61	42.04	33.25	7.01	27.13

Table 5. The main objectives of SimDE and its different variants for domain expansion.

Method	G_1 max	G_1 min	G_2 max	G_2 min
Variant A	$\mathcal{L}^1_{ent}$	$\mathcal{L}^1_{ce}$	N/A	N/A
Variant B	$\mathcal{L}^1_{ent}$	$\mathcal{L}^1_{ce}$	$\mathcal{L}^2_{ent}$	$\mathcal{L}^2_{ce}$
Variant C	$\mathcal{L}^1_{ent}$	$\mathcal{L}^1_{ce}$	$\mathcal{L}^1_{ent}$	$\mathcal{L}^1_{ce}$
Variant D	$\mathcal{L}^1_{ent} + \mathcal{L}^2_{ent}$	$\mathcal{L}^1_{ce} + \mathcal{L}^2_{ce}$	N/A	N/A
SimDE (ours)	$\mathcal{L}^2_{ent}$	$\mathcal{L}^1_{ce}$	$\mathcal{L}^1_{ent}$	$\mathcal{L}^2_{ce}$

4.3. Additional Analysis

Different choices of domain expansion objectives: To show the rationale of our methodology design, we compare SimDE with its different variants in Figure 3 and Table 5. The performance comparisons are shown in Table 6. First of all, variant A learns a single branch of generator and classifier with entropy maximization and cross-entropy minimization. We notice that the performance obtained by variant A can already reach the state-of-the-arts, showing the effectiveness of the simplified objective. Secondly, variant B introduces another full branch of generator and classifier into variant A, which means generator G_1 and G_2 use differ-



(a) Variant A (b) Variant B (c) Variant C

(d) Variant D (e) SimDE (ours)

Figure 3. Illustration of different domain expansion learning objectives of SimDE and its variants. For clarity, only the different implementations of entropy loss and cross-entropy loss are shown, while other components maintain the same across different variants.

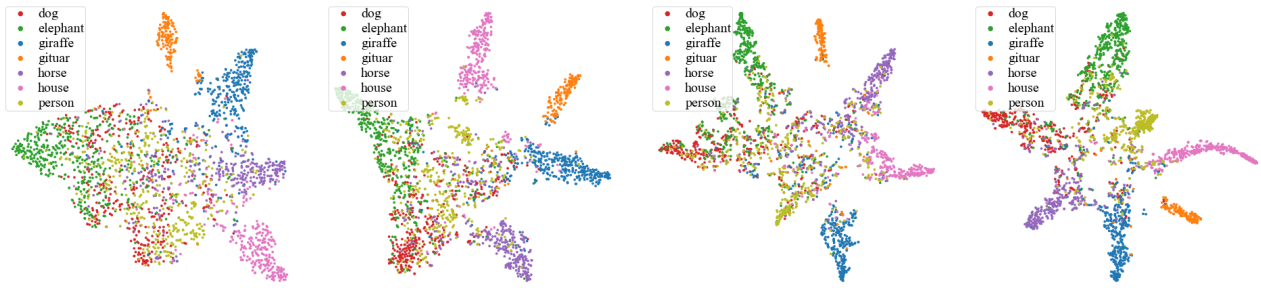
Table 6. Performances of different variants and loss combinations under single domain generalization on Digits and PACS. For each source domain, the averaged results on the target domains are reported. Note that $\mathcal{L}_{sty}$ and $\mathcal{L}_{con}$ are used in variants A∼D by default.

Method	Digits					PACS				
	SV	MM	SY	US	Avg	A	C	P	S	Avg
Variant A	64.07	81.83	66.26	84.18	74.09	77.87	75.45	56.60	55.10	66.26
Variant B	65.40	83.17	68.13	84.27	75.24	78.02	75.55	58.88	55.39	66.96
Variant C	62.28	83.39	65.52	84.47	73.91	78.34	75.47	56.30	55.35	66.37
Variant D	64.62	82.00	66.70	83.45	74.19	77.64	74.81	57.52	55.45	66.36
SimDE (ours)	**66.08**	**84.90**	**70.04**	**86.56**	**76.89**	**78.52**	**76.14**	**59.32**	**56.39**	**67.59**
SimDE *w/o* $\mathcal{L}_{sty}$	66.00	82.86	69.43	86.25	76.14	78.16	75.85	58.78	55.92	67.18
SimDE *w/o* $\mathcal{L}_{con}$	65.46	83.69	68.60	84.48	75.56	77.95	76.09	57.53	55.79	66.84

ent objectives. Such a simple extension can improve the final performances. However, the dual classifier discrepancy is not fully utilized in variant B and redundancy may still exists between the generations from G_1 and G_2. Thirdly, variant C only introduces another generator into variant A and the performance improvement is marginal, indicating that merely adding generators with the same objective is not beneficial. Similarly, variant D also fails to bring improvement by training a single generator with respect to the guidance of dual classifiers, indicating that better performance cannot be achieved with the collaborative training of two classifiers. Finally, our SimDE obtains the optimal performance by switching the objectives derived from dual classifiers, which could facilitate the generators to learn from different views as shown in Figure 2.

Ablation study on different components: We conduct ablation study about the style divergence loss $\mathcal{L}_{sty}$ and the contrastive loss $\mathcal{L}_{con}$ in our framework. As shown in Table 6, all these components make contributions to the final performance. Firstly, the contrastive loss brings consistent improvements in different cases, showing the advantage of enforcing feature invariance between domains. Secondly, although not crucial, the style divergence loss still boosts the overall performance, indicating that the generalizability can be improved by encouraging more styles in outputs from individual generators.

Choices of domain-invariant regularizations: The proposed SimDE is a general framework that can be incorporated with different domain alignment constraints. In Section 3.3, we instantiate the domain alignment in the manner of contrastive learning. Here we further show more instantiations. Specifically, classic discrepancy minimization loss like the MSE, JSD, or MMD [16] loss is included. Follow [41], we also develop a variant based on meta-learning by setting the original domain and the generated domain as meta-train and meta-test respectively. The results of different variants are shown in Table 7. In general, our method can benefit from different domain alignment



(a) ERM (b) MEADA (c) L2D (d) SimDE (ours)

Figure 4. The t-SNE visualizations of the target domain features with Skecth as the source domain on the PACS dataset. Features from the same class are plotted in the same color.

Table 7. Comparison of different domain-invariant regularizations on Digits dataset. Models are trained on MNIST. "SV", "MM", "SY", "US" denote SVHN, MNIST-M, SYN, USPS respectively.

Method	SV	MM	SY	US	Avg.
SimDE with JSD	65.76	84.32	67.89	86.18	76.04
SimDE with MSE	64.13	82.78	69.58	86.51	75.75
SimDE with MMD	**66.42**	83.92	69.70	84.75	76.20
SimDE with Meta	65.93	83.41	68.99	85.23	75.89
SimDE with CL (ours)	66.08	**84.90**	**70.04**	**86.56**	**76.89**

regularizations. Among all the variants, incorporating the JSD loss, the MMD loss and the contrastive loss generally brings better performances. The best overall performance is obtained by using the contrastive loss, suggesting its advantages in learning compact and invariant representations.

t-SNE visualizations of features: To further demonstrate the effectiveness of the proposed SimDE, we use t-SNE [46] to visualize the unseen target feature distribution of different methods in Figure 4. Specifically, we choose sketch on PACS dataset as the source domain since it has the largest domain shift. As shown in Figure 4, our method shows a better class separation than the competing method MEADA and L2D. It can be seen clearly that the feature distribution induced by our method is more compact and sparse, while other methods tend to mingle the features from different classes. This verifies the success of the domain expansion framework proposed by our method.

4.4. Extension to Few-shot Domain Adaptation

To further show the generalizability of our method, we conduct experiments under the few-shot domain adaptation setting [35], where a few labelled samples from the target domain as well as the whole source domain are used for training. Follow [29, 41], we use MNIST as the source domain and SVHN as the target. The models are first trained on the source domain with the proposed SimDE and then finetuned on the target domain. Models are evaluated on the test set of the target domain, as shown in Table 8. By finetuning with a few target samples, the model performance on target domain can be greatly improved, and saturates as the number of target samples increases. Our method consistently outperforms the domain expansion competitors M-ADA and PDEN in different cases of training samples. The results suggest that our method can induce a more unbiased model which benefits the downstream target adaptation.

Table 8. Few-shot domain adaptation accuracy with source domain as MNIST and target domain as SVHN.

Method	Training images per class		
	0	7	10
FADA [35]	-	47.00	-
CCSA [36]	-	-	37.63
M-ADA [41]	36.61	56.33	57.16
PDEN [29]	60.26	69.28	70.10
SimDE (ours)	**64.23**	**70.89**	**71.50**

5. Conclusion

In this paper, we propose a new and simple domain expansion learning objective for SingleDG by generating unconfident samples through the trade-off between entropy maximization and cross-entropy minimization. To enhance the diversity of domain expansions, we further propose a novel framework that trains a pair of generators by switching the guidance of dual classifiers. The proposed method, called *Simple Domain Expansion* (SimDE) forms a two-step adversarial training process, where the task models compete with the generators to improve the generalizability to unseen target domains. Extensive experiments on popular SingleDG benchmarks show that SimDE can reach state-of-the-arts results, demonstrating the success of the proposed domain expansion learning framework.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 1, 3

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

[3] Silvia Bucci, Antonio D'Innocente, Yujun Liao, Fabio M Carlucci, Barbara Caputo, and Tatiana Tommasi. Self-supervised learning across domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5516–5528, 2021. 1, 3

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2229–2238, 2019. 1, 3, 5, 6

[5] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pages 301–318. Springer, 2020. 3

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 5, 6

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5, 6

[8] Ilke Cugu, Massimiliano Mancini, Yanbei Chen, and Zeynep Akata. Attention consistency on visual corruptions for single-source domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4174, 2022. 3, 5, 6

[9] John Denker, W Gardner, Hans Graf, Donnie Henderson, R Howard, W Hubbard, Lawrence D Jackel, Henry Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. *Advances in neural information processing systems*, 1, 1988. 5

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 6

[11] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6450–6461, 2019. 1, 3

[12] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217, 2021. 3

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 3

[14] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 5

[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. 3

[16] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 7

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 5

[18] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. 6

[19] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017. 4

[20] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pages 124–140. Springer, 2020. 3, 6

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 4

[22] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 1

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 5

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

[25] 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. 1, 3

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1, 5

[27] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF Inter-*

national Conference on Computer Vision, pages 1446–1455, 2019. 3

[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*, pages 5400–5409, 2018. 1, 3

[29] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2021. 1, 2, 3, 4, 5, 6, 8

[30] 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, 3

[31] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019. 1, 3

[32] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 1

[33] Fabio Maria Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Hallucinating agnostic images to generalize across domains. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 0–0, 2019. 1, 3

[34] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11749–11756, 2020. 1, 3

[35] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, 30, 2017. 8

[36] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017. 1, 3, 5, 6, 8

[37] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021. 3, 6

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 5

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 5

[40] Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2021. 1, 2, 3, 5

[41] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 1, 3, 5, 6, 7, 8

[42] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018. 1, 3

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 3

[44] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 3

[45] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. 3

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[47] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018. 1, 3, 5, 6

[48] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1

[49] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 5

[50] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsicsupervisions for domain generalization. In *The European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[51] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021. 1, 2, 3, 4, 5, 6

[52] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. 1

[53] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021. 1, 3

[54] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2497–2506, 2019. 5, 6

[55] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 6

[56] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 3, 5, 6

[57] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020. 1, 3, 5, 6

[58] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032, 2020. 1, 3

[59] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 1, 3