Learning to Generate Novel Domains for Domain Generalization

Kaiyang Zhou[1], Yongxin Yang[1], Timothy Hospedales[2,3], and Tao Xiang[1,3]

[1] University of Surrey
{k.zhou, yongxin.yang, t.xiang}@surrey.ac.uk
[2] University of Edinburgh
t.hospedales@ed.ac.uk
[3] Samsung AI Center, Cambridge

Abstract. This paper focuses on domain generalization (DG), the task of learning from multiple source domains a model that generalizes well to unseen domains. A main challenge for DG is that the available source domains often exhibit limited diversity, hampering the model's ability to learn to generalize. We therefore employ a data generator to synthesize data from pseudo-novel domains to augment the source domains. This explicitly increases the diversity of available training domains and leads to a more generalizable model. To train the generator, we model the distribution divergence between source and synthesized pseudo-novel domains using optimal transport, and maximize the divergence. To ensure that semantics are preserved in the synthesized data, we further impose cycle-consistency and classification losses on the generator. Our method, L2A-OT (Learning to Augment by Optimal Transport) outperforms current state-of-the-art DG methods on four benchmark datasets.

1 Introduction

Humans effortlessly generalize prior knowledge to novel scenarios, a capability that machines still struggle to reproduce. Typically, machine-learning models perform poorly when deployed on test data with a different data distribution than the training data, which is known as the domain shift problem [35]. One line of research towards alleviating the domain shift problem is unsupervised domain adaptation (UDA), which exploits unlabeled target domain data for model adaptation [12, 33, 20, 40, 53, 44]. Although UDA methods avoid costly data annotation processes from target domains, data collection and per-domain model updates are still required. Meanwhile, UDA's assumption that target data can be collected in advance is not always met in practice [37, 10]. This motivates another line of research, namely domain generalization (DG) [37, 16, 15, 2, 5, 10], which is the main focus in this paper.

DG methods aim to learn models capable of good direct generalization to unseen target domains without data collection or model updating [37]. They usually, but not always [52], leverage multiple source domains to train a generalizable model. Most existing DG methods focus on aligning available source domains [36, 15, 11, 16, 29, 28], which is mainly inspired by UDA methods that seek

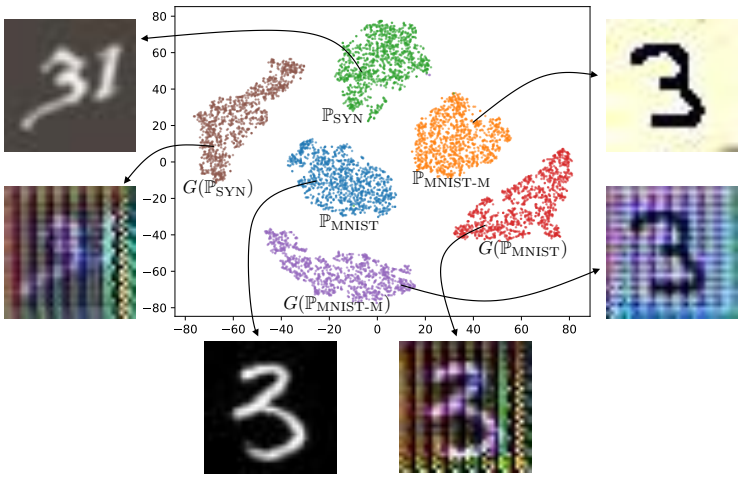

Fig. 1. Motivation of our approach. We improve generalization by increasing the diversity of training domains by learning a generator network G to map images of a source distribution, e.g., $\mathbb{P}_{\text{MNIST}}$, to a novel distribution, i.e. $G(\mathbb{P}_{\text{MNIST}})$. We then combine both source and novel domains for model learning.

to minimize the divergence between source data and unlabeled target data [13, 50]. As proved in [4], minimizing the domain divergence can lead to a smaller target error in the UDA setting. However, since DG methods focus on aligning source domains and do not have access to the target data, this theoretical proof does not apply to the DG setting. Recently, meta-learning has been exploited for DG where the key idea is to simulate domain shift by splitting the training data into meta-train and meta-test sets with non-overlapping domains [26, 2, 31, 27, 10]. During learning, models are optimized on the meta-train domains in a way that the error is reduced on the meta-test domains. Nevertheless, similar to the alignment-based methods, meta-learning optimizes for reducing the domain gap among source domains, and thus still has the risk of overfitting to seen domains.

In this paper, we address DG from a different perspective, i.e., the most straightforward way to improve model generalization is increasing the diversity of available source domains [49] (see Fig. 1). To this end, we propose *L2A-OT* (*Learning to Augment by Optimal Transport*). The core idea is to learn a conditional generator network that maps source domain images to pseudo-novel domains, and then combine both source and pseudo-novel domain images for training the actual task model. To train the generator, we *maximize* the distance between source domains and the generated pseudo-novel domains, as measured by optimal transport (OT) [41]. This leads to the generated images having a very different distribution from the source domains (Fig. 1). However, this objective alone does not guarantee that the semantic content of the generated images is preserved. Therefore, we further impose two losses on the generator, namely a

cycle-consistency loss [64] and a classification loss, for maintaining the structural and semantic consistency respectively.

Our contributions are as follows. **(1)** For the first time, DG is tackled from a perspective of pseudo-novel domain synthesis. **(2)** A novel image generator is formulated which differs from existing generators in the objective (synthesizing pseudo-novel domain images vs. natural photo images). More importantly it has a unique OT-based formulation of objective functions that allow the generator to explore novel domain space and generate diverse data with distributions different from any of the original source domains. We evaluate L2A-OT on three homogeneous DG benchmark datasets[4] including digit recognition [24, 12, 38], PACS [25] and Office-Home [51] and a heterogeneous DG task in the form of cross-domain person re-identification (re-ID) [58, 59, 32, 61, 22]. The results show that L2A-OT surpasses the current state-of-the-art on all datasets.

2 Related Work

Domain generalization. Many DG methods are based on the idea of domain alignment popularized from the UDA literature [12], with a goal to learn a domain-invariant representation by minimizing the domain discrepancy between sources [36, 15, 11, 16, 29, 28]. As mentioned earlier, aligning domain distributions is mainly motivated by the theory [4] developed for UDA, which does not apply to DG due to the absence of target data. Therefore, the models learned with domain alignment risk overfitting to source domains and as a result generalize poorly to unseen domains. In recent years, meta-learning [21] has seen increasing interest for DG where the objective is to expose a model to domain shift during training. This can be achieved by dividing source domains into meta-train and meta-test sets without overlapping, and training a model on the meta-train set such that the error on the meta-test set is reduced [26, 2, 10]. Similar to domain alignment methods, meta-learning methods still risk overfitting since the training data remains unchanged. Moreover, these methods work at feature-level, which is difficult for diagnosis and lacks visual interpretation.

Most related to our work are data augmentation methods, especially those based on adversarial gradients [47, 52]. For instance, [47] proposed CrossGrad to perturb input images with adversarial gradients generated by a domain classifier. Different from adversarial gradient-based methods which only produce imperceptible and simple pixel-wise effects (due to the nature of adversarial attack [48]), our approach *learns* a full CNN generator to map source images to unseen domains and optimizes it via OT-based distribution divergence to make the new domains as dissimilar as possible to source distributions.

Domain randomization. Our approach shares a similar high-level intuition with domain randomization (DR) [49], which was originally introduced in the context of robotic learning to improve generalization from simulation to real

[4] Following [31], homogeneous DG shares the same label space between training and test data while heterogeneous DG has disjoint label space.

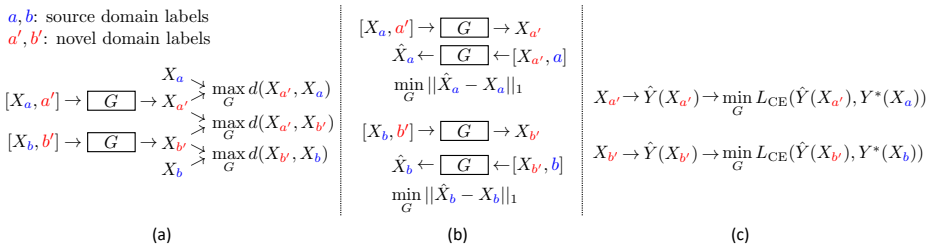

Fig. 2. Overview of our approach. (a) The conditional generator network G is learned to map input X to novel domains whose distributions are drastically different from the source domains, while keeping the distance between the novel domains as far as possible. (b) A cycle-consistency loss is imposed on G to maintain the structural consistency. (c) The cross-entropy loss is minimized with respect to G, using a pre-trained classifier $\hat{Y}$, for maintaining the semantic consistency.

world. DR aims to diversify the training domains by changing the color and texture of objects, background scenes, lighting conditions, etc. via a computer simulator [49]. Recently, DR has been successfully used in some computer vision applications, such as semantic segmentation [54, 55] and vehicle detection for autonomous driving [42]. However, our approach is significantly different from the DR-based methods because we *learn* a CNN generator network from real images rather than using programmatic simulators. Thus our method is more scalable to a wider range of image recognition tasks.

Image-to-image translation. Our work is also related to multi-domain image-to-image translation methods such as CycleGAN [64] and StarGAN [6], which use GAN losses [17] to generate realistic images and cycle-consistency losses [64] to achieve translation without using paired training images. Our method is fundamentally different from CycleGAN/StarGAN in that our generator model is learned to map source images to *unseen* domains rather than performing mapping between source domains as did in CycleGAN/StarGAN. We show by experiments that simply doing source-to-source mapping for data augmentation offers little help to DG (see Fig. 5a).

3 Methodology

3.1 Generating Novel-Domain Data

Setup. We are provided with K_s source domains with indices $D_s = \{1, 2, ..., K_s\}$. The goal is to learn a model which can generalize well on an unseen target domain. Without having access to the target data, we propose to improve the model's generalization by synthesizing novel data domains $D_n = \{1, 2, ..., K_n\}$ to augment the original source domains.

Conditional generator. We learn a conditional generator G (see Sec. 3.4 for detailed architecture design), that maps a source distribution $\mathbb{P}_k$ with $k \in D_s$ to a novel distribution $\mathbb{P}_{\tilde{k}}$ with $\tilde{k} \in D_n$ by conditioning on the novel domain label $\tilde{k}$, i.e. $\mathbb{P}_{\tilde{k}} = G(\mathbb{P}_k, \tilde{k})$. Here $\mathbb{P}$ denotes an empirical distribution rather than the real distribution, which is inaccessible. In practice, we use sampled mini-batches X_k instead of the full empirical distribution $\mathbb{P}_k$. Therefore, the domain translation function is defined as:

$$X_{\tilde{k}} = G(X_k, \tilde{k}). \tag{1}$$

Objective functions. For each training iteration, we randomly sample for each source domain k a mini-batch X_k, which is transformed to a randomly selected novel domain $\tilde{k} \sim D_n$. The objective is to force the novel distribution to be as dissimilar as possible to any source distribution, thus creating new domains to augment the existing source domains. We have

$$\max_G L_{\text{Novel}} = d(G(X_k, \tilde{k}), X_k), \tag{2}$$

where $d(\cdot, \cdot)$ is a distribution divergence measure (its design will be detailed in Sec. 3.5). Note that Eq. (2) will be summed over all source domains k, and each independently draws a novel domain label $\tilde{k}$.

In addition to maximizing the difference between source and novel distributions, we also maximize the difference between the generated novel distributions, i.e.

$$\max_G L_{\text{Diversity}} = d(X_{\tilde{k}_1}, X_{\tilde{k}_2}), \tag{3}$$

where $\tilde{k}_1, \tilde{k}_2 \in D_n$ and $\tilde{k}_1 \neq \tilde{k}_2$. Eq. (3) is summed over all possible pairs of novel distributions generated in one iteration. This diversity constraint diversifies the generated distributions, ensuring that the model benefits from generating $K_n > 1$ novel distributions. It is analogous to the diversity term in some image generation tasks, such as style transfer [30] where the pixel/feature difference between style-transferred instances is maximized. Differently, our formulation focuses on the divergence between data distributions. See Fig. 2a for a graphical illustration.

3.2 Maintaining Semantic Consistency

The model so far is optimizing a powerful CNN generator G for the novelty of the generated distribution (Eq. (2) & (3)). This produces diverse images, but may not preserve their semantic content.

Cycle-consistency loss. First, to guarantee structural consistency, we apply a cycle-consistency constraint [64] to the generator,

$$\min_G L_{\text{Cycle}} = ||G(G(X_k, \tilde{k}), k) - X_k||_1, \tag{4}$$

where the outer G aims to reconstruct the original X_k given as input the domain-translated $G(X_k, \tilde{k})$ and the original domain label k. Both G's in the cycle share the same parameters [6]. This is illustrated in Fig. 2b.

Cross-entropy loss. Second, to maintain the category label and thus enforce semantic consistency, we further require that the generated data $X_{\tilde{k}}$ is classified into the same category as the original data X_k, i.e.

$$\min_G L_{\text{CE}}(\hat{Y}(X_{\tilde{k}}), Y^*(X_k)), \tag{5}$$

where L_{CE} denotes cross-entropy loss, $\hat{Y}(X_{\tilde{k}})$ the labels of $X_{\tilde{k}}$ predicted by a pretrained classifier and $Y^*(X_k)$ the ground-truth labels of X_k. This is illustrated in Fig. 2c.

3.3 Training

Generator training. The full objective for G is the weighted combination of Eq. (2), (3), (4), & (5),

$$\begin{aligned}\min_G L_G = &- \lambda_{\text{Domain}}(L_{\text{Novel}} + L_{\text{Diversity}}) \\ &+ \lambda_{\text{Cycle}} L_{\text{Cycle}} + \lambda_{\text{CE}} L_{\text{CE}},\end{aligned} \tag{6}$$

where λ_{Domain}, λ_{Cycle} and λ_{CE} are weighting hyper-parameters.

Task model training. The task model F is trained from scratch using both the original data X_k and the synthetic data $X_{\tilde{k}}$ generated as described above. The objective for F is

$$\min_F L_F = (1-\alpha) L_{\text{CE}} + \alpha \tilde{L}_{\text{CE}}, \tag{7}$$

where α is a balancing weight, which is fixed to 0.5 throughout this paper; L_{CE} and $\tilde{L}_{\text{CE}}$ are the cross-entropy losses computed using X_k and $X_{\tilde{k}}$ respectively. The full training algorithm is shown in Alg. **??** (In the Supp.). Note that each source domain $k \in D_s$ will be assigned a unique novel domain $\tilde{k} \in D_n$ as target in each iteration. We set $K_n = K_s$ as default.

3.4 Design of Conditional Generator Network

Our generator model has a conv-deconv structure [64, 6] which is shown in Fig. 3. Specifically, the generator model consists of two down-sampling convolution layers with stride 2, two residual blocks [19] and two transposed convolution layers with stride 2 for up-sampling. Following StarGAN [6], the domain indicator is encoded as a one-hot vector with length $K_s + K_n$ (see Fig. 3). During the forward pass, the one-hot vector is first spatially expanded and then concatenated with the image to form the input to G.

Discussion. Though the design of G is similar to the StarGAN model, their learning objectives are totally different: We aim to generate images that are different from the existing source domain distributions while the StarGAN model is trained to generate images from the existing source domains. In the experiment part we justify that adding novel-domain data is much more effective than adding seen-domain data for DG (see Fig. 5a). Compared with the gradient-based perturbation method in [47], our generator is allowed to model more sophisticated domain shift such as image style changes due to its learnable nature.

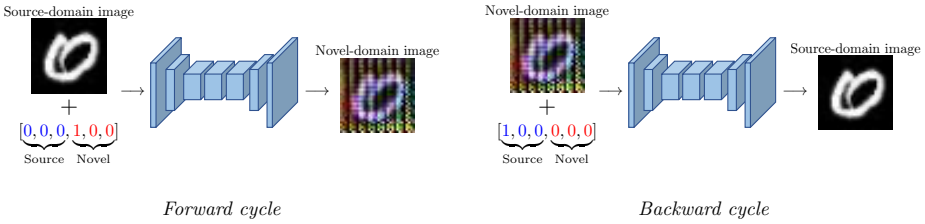

Fig. 3. Architecture of the conditional generator network. Left and right images exemplify the forward cycle and backward cycle respectively in cycle-consistency.

3.5 Design of Distribution Divergence Measure

Two common families for estimating the divergence between probability distributions are f-divergence (e.g., KL divergence) and integral probability metrics (e.g., Wasserstein distance). In contrast to most work that minimizes the divergence, we need to maximize it, as shown in Eq. (2) & (3). This strongly suggests to avoid f-divergence because of the near-zero denominators (they tend to generate large but numerically unstable divergence values). Therefore, we choose the second type, specifically the Wasserstein distance, which has been widely used in recent generative modeling methods [1, 14, 3, 45, 46].

The Wasserstein distance, also known as optimal transport (OT) distance, is defined as

$$\mathcal{W}_c(\mathbb{P}_a, \mathbb{P}_b) = \inf_{\pi \in \Pi(\mathbb{P}_a, \mathbb{P}_b)} \mathbb{E}_{x_a, x_b \sim \pi}[c(x_a, x_b)], \tag{8}$$

where $\Pi(\mathbb{P}_a, \mathbb{P}_b)$ denotes the set of all joint distributions $\pi(x_a, x_b)$ and $c(\cdot, \cdot)$ the transport cost function. Intuitively, the OT metric computes the minimum cost of transporting masses between distributions in order to turn $\mathbb{P}_b$ into $\mathbb{P}_a$.

As the sampling over $\Pi(\mathbb{P}_a, \mathbb{P}_b)$ is intractable, we resort to using the entropy-regularized Sinkhorn distance [7]. Moreover, to obtain unbiased gradient estimators when using mini-batches, we adopt the generalized (squared) energy distance [45], leading to

$$d(\mathbb{P}_a, \mathbb{P}_b) = 2\mathbb{E}[\mathcal{W}_c(X_a, X_b)] - \mathbb{E}[\mathcal{W}_c(X_a, X'_a)] - \mathbb{E}[\mathcal{W}_c(X_b, X'_b)], \tag{9}$$

where X_a and X'_a are independent mini-batches from distribution $\mathbb{P}_a$; X_b and X'_b are independent mini-batches from distribution $\mathbb{P}_b$; $\mathcal{W}_c$ is the Sinkhorn distance defined as

$$\mathcal{W}_c(\cdot, \cdot) = \inf_{M \in \mathcal{M}} \sum_{i,j} [M \odot C]_{i,j}, \tag{10}$$

where the soft-matching matrix M represents the coupling distribution π in Eq. (8) and can be efficiently computed using the Sinkhorn algorithm [14]; C is the pairwise distance matrix computed over two sets of samples.

Following [45], we define the cost function as the cosine distance between instances,

$$c(x_a, x_b) = 1 - \frac{\phi(x_a)^T \phi(x_b)}{||\phi(x_a)||_2 ||\phi(x_b)||_2}, \tag{11}$$

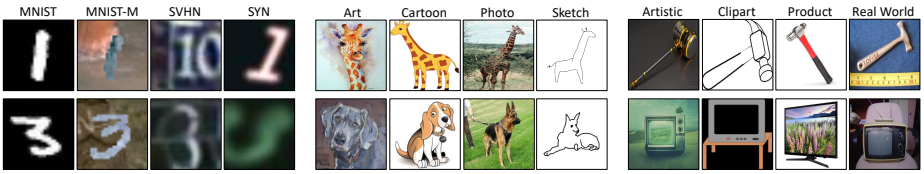

Fig. 4. Example images from different DG datasets.

where ϕ is constructed by a CNN (also called critic in [45]), which maps images into a latent space. In practice, ϕ is a fixed CNN that was trained with domain classification loss.

4 Experiments

4.1 Evaluation on Homogeneous DG

Datasets. (1) We use four different digit datasets including MNIST [24], MNIST-M [12], SVHN [38] and SYN [12], which differ drastically in font style, stroke color and background. We call this new dataset **Digits-DG** hereafter. See Fig. 4a for example images. (2) **PACS** [25] is composed of four domains, which are Photo, Art Painting, Cartoon and Sketch, with 9,991 images of 7 classes in total. See Fig. 4b for example images. (3) **Office-Home** [51] contains around 15,500 images of 65 classes for object recognition in office and home environments. It has four domains, which are Artistic, Clipart, Product and Real World. See Fig. 4c for example images.

Evaluation protocol. For fair comparison with prior work, we follow the leave-one-domain-out protocol in [25, 5, 27]. Specifically, one domain is chosen as the test domain while the remaining domains are used as source domains for model training. The top-1 classification accuracy is used as performance measure. All results are averaged over three runs with different random seeds.

Baselines. We compare L2A-OT with the recent state-of-the-art DG methods that report results on the same dataset or have code publicly available for reproduction. These include (1) **CrossGrad** [47], the most related work that perturbs input using adversarial gradients from a domain classifier; (2) **CCSA** [36], which learns a domain-invariant representation using a contrastive semantic alignment loss; (3) **MMD-AAE** [28], which imposes a MMD loss on the hidden layers of an autoencoder. (4) **JiGen** [5], which has an auxiliary self-supervision loss to solve the Jigsaw puzzle task [39]; (5) **Epi-FCR** [27], which designs an episodic training strategy; (6) A **vanilla** model trained by aggregating all source domains, which serves as a strong baseline.

Method	MNIST	MNIST-M	SVHN	SYN	Avg.
Vanilla	95.8	58.8	61.7	78.6	73.7
CCSA [36]	95.2	58.2	65.5	79.1	74.5
MMD-AAE [28]	96.5	58.4	65.0	78.4	74.6
CrossGrad [47]	**96.7**	61.1	65.3	80.2	75.8
JiGen [5]	96.5	61.4	63.7	74.0	73.9
L2A-OT (*ours*)	**96.7**	**63.9**	**68.6**	**83.2**	**78.1**

Table 1. Leave-one-domain-out results on Digits-DG.

Implementation details. For Digits-DG, the CNN backbone is constructed with four 64-kernel 3×3 convolution layers and a softmax layer. ReLU and 2×2 max-pooling are inserted after each convolution layer. F is trained with SGD, initial learning rate of 0.05 and batch size of 126 (42 images per source) for 50 epochs. The learning rate is decayed by 0.1 every 20 epochs. For all experiments, G is trained with Adam [23] and a constant learning rate of 0.0003. For both PACS and Office-Home, we use ResNet-18 [19] pretrained on ImageNet [8] as the CNN backbone, following [9, 5, 27]. On PACS, F is trained with SGD, initial learning rate of 0.00065 and batch size of 24 (8 images per source) for 40 epochs. The learning rate is decayed by 0.1 after 30 epochs. On Office-Home, the optimization parameters are similar to those on PACS except that the maximum epoch is 25 and the learning rate decay step is 20. For all datasets, as target data is unavailable during training, the values of hyper-parameters λ_{Domain}, λ_{Cycle} and λ_{CE} are set based on the performance on source validation set,[5] which is a strategy commonly adopted in the DG literature [5, 27]. Our implementation is based on `Dassl.pytorch` [63].

Results on Digits-DG. Table 1 shows that L2A-OT achieves the best performance on all domains and consistently outperforms the vanilla baseline by a large margin. Compared with CrossGrad, L2A-OT performs clearly better on MNIST-M, SVHN and SYN, with clear improvements of 2.8%, 3.3% and 3%, respectively. It is worth noting that these three domains are very challenging with large domain variations compared with their source domains (see Fig. 4a). The huge advantage over CrossGrad can be attributed to L2A-OT's unique generation of unseen-domain data using a fully learnable CNN generator, and using optimal transport to explicitly encourage domain divergence. Compared with the domain alignment methods, L2A-OT surpasses MMD-AAE and CCSA by more than 3.5% on average. The is because L2A-OT enriches the domain diversity of training data, thus reducing overfitting in source domains. L2A-OT clearly beats JiGen because the Jigsaw puzzle transformation does not work well on digit images with sparse pixels [39].

Results on PACS. The results are shown in Table 2. Overall, L2A-OT achieves the best performance on all test domains. L2A-OT clearly beats the latest DG methods, JiGen and Epi-FCR. This is because our classifier benefits from the

[5] The searching space is: $\lambda_{\text{Domain}} \in \{0.5, 1, 2\}$, $\lambda_{\text{Cycle}} \in \{10, 20\}$ and $\lambda_{\text{CE}} \in \{1\}$.

Method	Art	Cartoon	Photo	Sketch	Avg.
Vanilla	77.0	75.9	96.0	69.2	79.5
CCSA [36]	80.5	76.9	93.6	66.8	79.4
MMD-AAE [28]	75.2	72.7	96.0	64.2	77.0
CrossGrad [47]	79.8	76.8	96.0	70.2	80.7
JiGen [5]	79.4	75.3	96.0	71.6	80.5
Epi-FCR [27]	82.1	77.0	93.9	73.0	81.5
L2A-OT (*ours*)	**83.3**	**78.2**	**96.2**	**73.6**	**82.8**

Table 2. Leave-one-domain-out results on PACS dataset.

Method	Artistic	Clipart	Product	Real World	Avg.
Vanilla	58.9	49.4	74.3	76.2	64.7
CCSA [36]	59.9	49.9	74.1	75.7	64.9
MMD-AAE [28]	56.5	47.3	72.1	74.8	62.7
CrossGrad [47]	58.4	49.4	73.9	75.8	64.4
JiGen [5]	53.0	47.5	71.5	72.8	61.2
L2A-OT (*ours*)	**60.6**	**50.1**	**74.8**	**77.0**	**65.6**

Table 3. Leave-one-domain-out results on Office-Home.

generated unseen-domain data while JiGen and Epi-FCR, like the domain alignment methods, are prone to overfitting to the source domains. L2A-OT beats CrossGrad on all domains, mostly with a large margin. This again justifies our design of learnable CNN generator over adversarial gradient.

Results on Office-Home. The results are reported in Table 3. Again, L2A-OT achieves the best overall performance, and other conclusions drawn previously also hold. Notably, the simple vanilla model obtains strong results on this benchmark, which are even better than most existing DG methods. This is because the dataset is relatively large, and the domain shift is less severe compared with the style changes on PACS and the font variations on Digits-DG.

4.2 Evaluation on Heterogeneous DG

In this section, we evaluate L2A-OT on a more challenging DG task with disjoint label space between training and test data, namely cross-domain person re-identification (re-ID).

Datasets. We use Market1501 [56] and DukeMTMC-reID (Duke) [43, 57]. Market1501 has 32,668 images of 1,501 identities captured by 6 cameras (domains). Duke has 36,411 images of 1,812 identities captured by 8 cameras.

Evaluation protocol. We follow the recent unsupervised domain adaptation (UDA) methods in the person re-ID literature [58, 59, 32] and experiment with Market1501→Duke and Duke→Market1501. Different from the UDA setting, we directly test the source-trained model on the target dataset without adaptation. Note that the cross-domain re-ID evaluation involves training a person classifier on source dataset identities. This is then transferred and used to recognize a

Method	Market1501→Duke				Duke→Market1501			
	mAP	R1	R5	R10	mAP	R1	R5	R10
UDA methods								
ATNet [32]	24.9	45.1	59.5	64.2	25.6	55.7	73.2	79.4
CamStyle [59]	25.1	**48.4**	**62.5**	**68.9**	27.4	58.8	78.2	**84.3**
HHL [58]	**27.2**	46.9	61.0	66.7	**31.4**	**62.2**	**78.8**	84.0
DG methods								
Vanilla	26.7	48.5	62.3	67.4	26.1	57.7	73.7	80.0
CrossGrad [47]	27.1	48.5	63.5	69.5	26.3	56.7	73.5	79.5
L2A-OT (*ours*)	**29.2**	**50.1**	**64.5**	**70.1**	**30.2**	**63.8**	**80.2**	**84.6**

Table 4. Results on cross-domain person re-ID benchmarks.

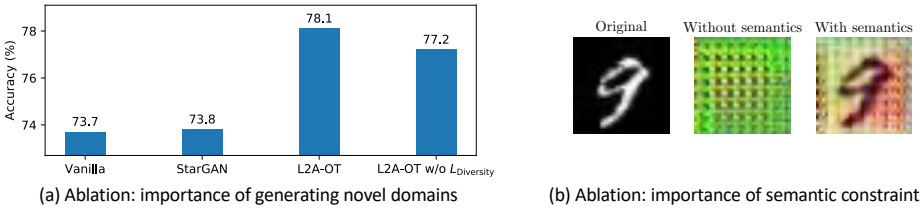

Fig. 5. Ablation study.

disjoint set of people in the target domain of unseen camera views via nearest neighbor. Since the label space is disjoint, this is a *heterogeneous* DG problem. For performance measure, we adopt CMC ranks and mAP [56].

Implementation details. For the CNN backbone, we employ the state-of-the-art re-ID model, OSNet-IBN [62, 61]. Following [62, 61], OSNet-IBN is trained using the standard classification paradigm, i.e. each identity is considered as a class. Therefore, the entire L2A-OT framework remains unchanged. At test time, feature vectors extracted from OSNet-IBN are used to compute ℓ_2 distance for image matching. Our implementation is based on `Torchreid` [60].

Results. In Table 4, we compare L2A-OT with the vanilla model and CrossGrad, as well as state-of-the-art UDA methods for re-ID. As a result, CrossGrad barely improves the vanilla model while L2A-OT achieves clear improvements on both settings. Notably, L2A-OT is highly competitive with the UDA methods, though the latter make the significantly stronger assumption of having access to the target domain data (thus gaining an unfair advantage). In contrast, L2A-OT generates images of unseen styles (domains) for data augmentation, and such more diverse data leads to learning a better generalizable re-ID model.

4.3 Ablation Study

Importance of generating novel domains. To verify that our improvement is brought by the increase in training data distributions by the generated novel domains (i.e. Eq. (2) & (3)), we compare L2A-OT with StarGAN [6], which

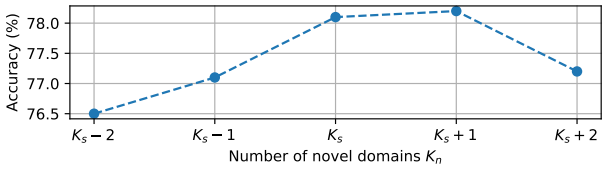

Fig. 6. Results of varying K_n. Here $K_s = 3$.

Source			Target	L2A-OT	Vanilla
MNIST	SVHN	SYN			
✓	✓		MNIST-M	60.9	54.6
✓		✓	MNIST-M	62.1	**59.1**
	✓	✓	MNIST-M	49.7	45.2
✓	✓	✓	MNIST-M	**62.5**	57.1

Table 5. Using two vs. three source domains on Digits-DG where the size of training data is kept identical for all settings for fair comparison.

generates data from the existing source domains by performing source-to-source mapping. The experiment is conducted on Digits-DG and the average performance over all test domains is used for comparison. Fig. 5a shows that StarGAN performs only similarly to the vanilla model (StarGAN's 73.8% vs. vanilla's 73.7%) while L2A-OT obtains a clear improvement of 4.3% over StarGAN. This confirms that increasing domains is far more important than increasing data (of seen domains) for DG. Note that this 4.3% gap is attributed to the combination of the OT-based domain novelty loss (Eq. (2)) and the diversity loss (Eq. (3)). Fig. 5a shows that the diversity loss contributes around 1% to the performance, and the rest improvement comes from the diversity loss.

Importance of semantic constraint. The cycle-consistency and cross-entropy losses (Eq. (4) & (5)) are essential in the L2A-OT framework for maintaining the semantic content when performing domain translation. Fig. 5b shows that without the semantic constraint, the content is completely missing (we found that using these images reduced the result from 78.1% to 73.9%).

4.4 Further Analysis

How many novel domains to generate? Our approach can generate an arbitrary number of novel domains, although we have always doubled the number of domains (set $K_s = K_n$) so far. Fig. 6 investigates the significance on the choice of number of novel domains. In principle, synthesizing more domains provides opportunity for more diverse data, but also increases optimization difficulty and is dependent on the source domains. The result shows that the performance is not very sensitive to the choice of novel domain number, with $K_n = K_s$ being a good rule of thumb.

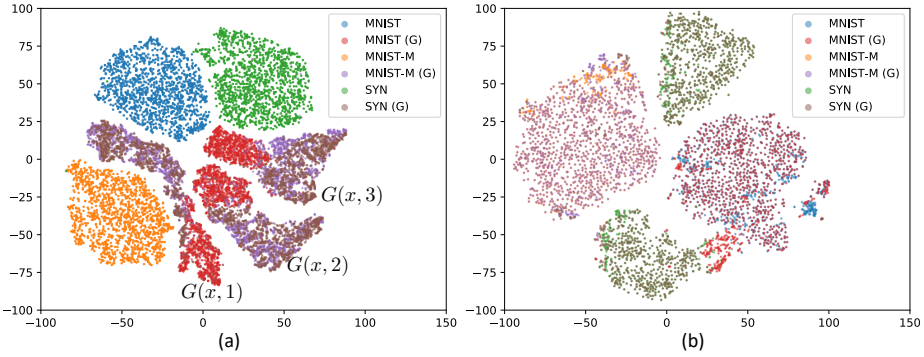

Fig. 7. T-SNE visualization of domain embeddings of (a) L2A-OT and (b) CrossGrad [47]. X (G) indicates novel data when using the domain X as a source.

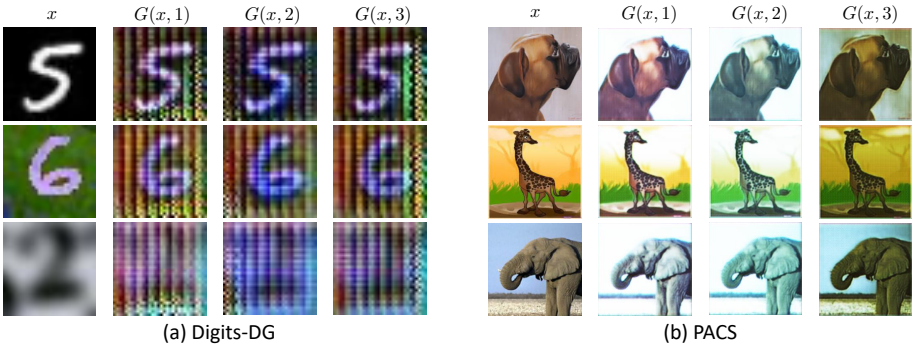

Fig. 8. Visualization of generated images. x: source image. $G(x,i)$: generated image of the i-th novel domain.

Do more source domains lead to a better result? In general, yes. The evidence is shown in Table 5 where the result of using three sources is generally better than using two as we might expect due to the additional diversity. The detailed results show that when using two sources, performance is sensitive to the choice of sources among the available three. This is expected since different sources will vary in transferrability to a given target. However, for both vanilla and L2A-OT the performance of using three sources is better than the performance of using two averaged across the 2-source choices.

Visualizing domain distributions. We employ t-SNE [34] to visualize the domain feature embeddings using the validation set of Digits-DG (see Fig. 7a). We have the following observations. (1) The generated distributions are clearly separated from the source domains and evenly fill the unseen domain space. (2) The generated distributions form independent clusters (due to our diversity term in Eq. (3)). (3) G has successfully learned to flexibly transform one source domain to any of the discovered novel domains.

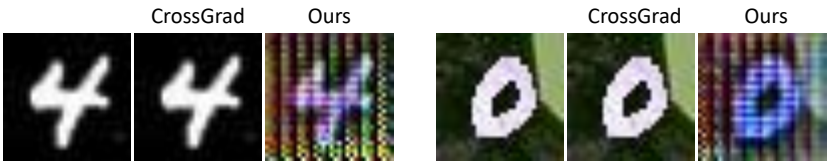

Fig. 9. Comparison between L2A-OT and CrossGrad [47] on image generation.

Visualizing novel-domain images. Fig. 8 visualizes the output of G. In general, we observe that the generated images from different novel domains manifest different properties and more importantly, are clearly different from the source images. For example, in Digits-DG (Fig. 8a), G tends to generate images with different background patterns/textures and font colors. In PACS (Fig. 8b), G focuses on contrast and color. Fig. 8 seems to suggest that the synthesized domains are not drastically different from each other. However, a seemingly limited diversity in the image space to human eyes can be significant to a CNN classifier: both Fig. 1 and Fig. 7a show clearly that the synthesized data points have very different distributions from both the original ones and each other in a feature embedding space, making them useful for learning a domain-generalizable classifier.

L2A-OT vs. CrossGrad. It is clear from Fig. 7b that the new domains generated by CrossGrad largely overlap with the original domains. This is because CrossGrad is based on adversarial attack methods [18], which are designed to make imperceptible changes. This is further verified in Fig. 9 where the images generated by CrossGrad have only subtle differences in contrast to the original images. On the contrary, L2A-OT can model much more complex domain variations that can materially benefit the classifier, thanks to the full CNN image generator and OT-based domain divergence losses.

5 Conclusion

We presented L2A-OT, a novel data augmentation-based DG method that boosts classifier's robustness to domain shift by learning to synthesize images from diverse unseen domains through a conditional generator network. The generator is trained by maximizing the OT distance between source domains and pseudo-novel domains. Cycle-consistency and classification losses are imposed on the generator to further maintain the structural and semantic consistency during domain translation. Extensive experiments on four DG benchmark datasets covering a wide range of visual recognition tasks demonstrate the effectiveness and versatility of L2A-OT.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017)
2. Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization. In: NeurIPS (2018)
3. Bellemare, M.G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., Munos, R.: The cramer distance as a solution to biased wasserstein gradients. arXiv preprint arXiv:1705.10743 (2017)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. ML (2010)
5. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: CVPR (2019)
6. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR (2018)
7. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NeurIPS (2013)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
9. D'Innocente, A., Caputo, B.: Domain generalization with domain-specific aggregation modules. In: GCPR (2018)
10. Dou, Q., Castro, D.C., Kamnitsas, K., Glocker, B.: Domain generalization via model-agnostic learning of semantic features. In: NeurIPS (2019)
11. Gan, C., Yang, T., Gong, B.: Learning attributes equals multi-source domain generalization. In: CVPR (2016)
12. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
13. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR (2016)
14. Genevay, A., Peyré, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: AISTATS (2018)
15. Ghifary, M., Balduzzi, D., Kleijn, W.B., Zhang, M.: Scatter component analysis: A unified framework for domain adaptation and domain generalization. TPAMI (2017)
16. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: ICCV (2015)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
20. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
21. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. arXiv preprint arXiv:2004.05439 (2020)
22. Jin, X., Lan, C., Zeng, W., Chen, Z., Zhang, L.: Style normalization and restitution for generalizable person re-identification. In: CVPR (2020)

23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2014)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: IEEE (1998)
25. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: ICCV (2017)
26. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization. In: AAAI (2018)
27. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.M.: Episodic training for domain generalization. In: ICCV (2019)
28. Li, H., Jialin Pan, S., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: CVPR (2018)
29. Li, Y., Tiana, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D.: Deep domain generalization via conditional invariant adversarial networks. In: ECCV (2018)
30. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: CVPR (2017)
31. Li, Y., Yang, Y., Zhou, W., Hospedales, T.: Feature-critic networks for heterogeneous domain generalization. In: ICML (2019)
32. Liu, J., Zha, Z.J., Chen, D., Hong, R., Wang, M.: Adaptive transfer network for cross-domain person re-identification. In: CVPR (2019)
33. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML (2015)
34. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
35. Moreno-Torres, J.G., Raeder, T., Alaiz-RodríGuez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. PR (2012)
36. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: ICCV (2017)
37. Muandet, K., Balduzzi, D., Scholkopf, B.: Domain generalization via invariant feature representation. In: ICML (2013)
38. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NeurIPS-W (2011)
39. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
40. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV (2019)
41. Peyré, G., Cuturi, M., et al.: Computational optimal transport. Foundations and Trends® in Machine Learning (2019)
42. Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., Birchfield, S.: Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In: ICRA (2019)
43. Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV-W (2016)
44. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: ICCV (2019)
45. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving gans using optimal transport. In: ICLR (2018)
46. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: ICCV (2019)
47. Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., Sarawagi, S.: Generalizing across domains via cross-gradient training. In: ICLR (2018)

48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
49. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS (2017)
50. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)
51. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR (2017)
52. Volpi, R., Namkoong, H., Sener, O., Duchi, J., Murino, V., Savarese, S.: Generalizing to unseen domains via adversarial data augmentation. In: NeurIPS (2018)
53. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: ICCV (2019)
54. Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In: ICCV (2019)
55. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: ICCV (2019)
56. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: ICCV (2015)
57. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In: ICCV (2017)
58. Zhong, Z., Zheng, L., Li, S., Yang, Y.: Generalizing a person retrieval model hetero- and homogeneously. In: ECCV (2018)
59. Zhong, Z., Zheng, L., Zheng, Z., Li, S., Yang, Y.: Camstyle: A novel data augmentation method for person re-identification. TIP (2019)
60. Zhou, K., Xiang, T.: Torchreid: A library for deep learning person re-identification in pytorch. arXiv preprint arXiv:1910.10093 (2019)
61. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning generalisable omni-scale representations for person re-identification. arXiv preprint arXiv:1910.06827 (2019)
62. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Omni-scale feature learning for person re-identification. In: ICCV (2019)
63. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain adaptive ensemble learning. arXiv preprint arXiv:2003.07325 (2020)
64. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)