Domain Shift Preservation for Zero-Shot Domain Adaptation

Jinghua Wang, Ming-Ming Cheng, and Jianmin Jiang*

***Abstract*—In learning-based image processing a model that is learned in one domain often performs poorly in another since the image samples originate from different sources and thus have different distributions. Domain adaptation techniques alleviate the problem of domain shift by learning transferable knowledge from the source domain to the target domain. Zero-shot domain adaptation (ZSDA) refers to a category of challenging tasks in which no target-domain sample for the task of interest is accessible for training. To address this challenge, we propose a simple but effective method that is based on the strategy of domain shift preservation across tasks. First, we learn the shift between the source domain and the target domain from an irrelevant task for which sufficient data samples from both domains are available. Then, we transfer the domain shift to the task of interest under the hypothesis that different tasks may share the domain shift for a specified pair of domains. Via this strategy, we can learn a model for the unseen target domain of the task of interest. Our method uses two coupled generative adversarial networks (CoGANs) to capture the joint distribution of data samples in dual-domains and another generative adversarial network (GAN) to explicitly model the domain shift. The experimental results on image classification and semantic segmentation demonstrate the satisfactory performance of our method in transferring various kinds of domain shifts across tasks.**

***Index Terms*—Domain adaptation, zero-shot domain adaptation, zero-shot learning, coupled generative adversarial networks, adversarial learning**

I. Introduction

DOMAIN adaptation techniques alleviate the domain shift problem by learning a transferable model from the source domain to the target domain [1], [2]. While the source domain typically has many labeled data samples, the target domain often lacks labels or data samples. These techniques are successfully applied in various applications, such as semantic segmentation [3], [4], visual recognition [5], [6], action recognition [7], [8], and person re-identification [9], [10].

Domain adaptation tasks can be classified into four categories based on the available information (including data and labels) in the target domain, namely, supervised domain adaptation [11], semi-supervised domain adaptation [12], unsupervised domain adaptation [13], [14], and zero-shot domain adaptation (ZSDA) [15]–[20], as presented in Tab. I. While the target-domain data are available in the training procedure of the first three categories, the ZSDA tasks aim at learning a model for the unseen target domain. ZSDA is also referred to as domain generalization (DG) [17], [18], [21], [22]. A typical ZSDA task is the personalization of a new portable device, where we expect to learn a model before the user's data (the target-domain data) are provided. In this task, a domain refers to the settings and preferences of a user. We may also expect to learn a computer vision model for a new camera to enable operation immediately after installation. This involves a ZSDA task in which the target-domain data are the non-available images that are captured by the new camera [17] and the source-domain data are the images that are captured by the original camera.

TABLE I: Depending on whether labeled and unlabeled data are available in the target domain (T-Domain), domain adaptation tasks can be classified into four categories: supervised, semi-supervised, unsupervised and ZSDA.

T-Domain	Superv.	Semi-S.	Unsuperv.	ZSDA
Labeled	✓	✓	×	×
Unlabeled	×	✓	✓	×

Due to the non-availability of the target-domain data, ZSDA tasks are more challenging than the domain adaptation tasks of the other three types. To address ZSDA tasks, researchers either learn domain-invariant features [23]–[28] to minimize the domain shift or train domain agnostic models based on the common property among domains [29], [30]. However, the application scope of these methods [23]–[30] is limited by a common implicit assumption, namely, that the data samples from multiple source domains are available in the training stage. To learn a target-domain model with a single source domain, Peng *et al.* [15] propose borrowing knowledge from an irrelevant task for which data samples from both domains are available. Inspired by [15], our method also involves two tasks: the task of interest, namely, the relevant task (RT), and an irrelevant task (IRT). While only source domain data are available for the RT, data for both domains are available for the IRT. The proposed method aims at learning a model for the unseen target domain in the RT based on the source-domain data in the RT and the dual-domain data in the IRT.

Fig. 1 illustrates an example in which ZSDA learns a model for *MNIST-M* [1] from the data of *MNIST* [31], *Fashion-MNIST* [32], and *Fashion-MNIST-M*. *MNIST-M* and *Fashion-MNIST-M* are the color versions of *MNIST* and *Fashion-MNIST*, respectively. While the source domain consists of gray-scale images, the target domain consists of color images. The RT and IRT classify digit images and fashion images,

*Corresponding author: Jianmin Jiang

J Wang and J Jiang are with the Research Institute for Future Media Computing, College of Computer Science and Software Engineering, and Guangdong Laboratory of Artificial Intelligence & Digital Economy (SZ), Shenzhen University, Shenzhen 518060. E-mail: wang.jh@szu.edu.cn; jianmin.jiang@szu.edu.cn

MM Cheng is with the College of Computer Science, Nankai University. Email: cmm@nankai.edu.cn

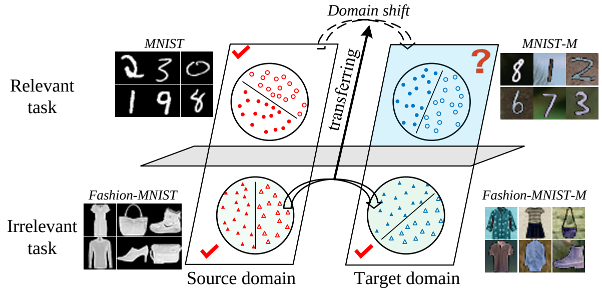


Fig. 1: Intuitive example of ZSDA (best viewed in color). For digit image analysis (the RT), it is difficult to learn a satisfactory model for the unseen *MNIST-M* based solely on the available *MNIST*. Our method learns the domain shift based on fashion analysis (*i.e.,* the IRT) and transfers the domain shift to the digit image analysis. Hence, the model for *MNIST-M* is determined by the available *MNIST* and the domain shift that is learned in the fashion analysis.

respectively. In principle, the differences between the gray-scale images and the color images in both tasks are induced by the same colorization procedure [1]. Based on this observation, we establish the following hypothesis: The domain shift, which intrinsically characterizes the divergence between two domains, is shared by both the RT and the IRT. Thus, we can learn the domain shift from one task and transfer it to the other.

To address the ZSDA problem effectively, we propose the use of two coupled generative adversarial networks (CoGANs) [33] to capture the correlation between a pair of domains and one generative adversarial network (GAN) [34] to model the domain shift. First, we train a *CoGAN-IR* for the IRT to model both the domain-specific low-level details and the domain-invariant high-level concepts. We realize two objectives with this *CoGAN-IR*: (i) learn the joint distribution of dual-domain samples based on the independent source samples and target samples in the IRT; and (ii) provide a data foundation for domain shift modeling by generating the *paired samples*. Two samples are *paired samples* if they represent the same entity in two domains, such as a color image and a depth image of the same scene. The *domain shift* introduces the difference between the *paired samples*. Thus, it is more effective to analyze the domain shift with the *paired samples*. Second, we use a GAN^{shift} to model the domain shift. Specifically, the GAN^{shift} captures the distribution of the element-wise difference between the *paired samples* in the representation space. Via this strategy, we encode the domain shift with a network structure and, hence, render it more convenient to transfer. Finally, we transfer the domain shift across tasks by enabling the *CoGAN-R*, which captures the joint distribution of dual-domain samples of the RT, to carry the domain shift that is encoded in GAN^{shift}. To realize this objective, we propose a method for *CoGAN-R* training that is based on the joint supervision of the source-domain data and the domain shift, which can guide the learning of the high-level concepts and the low-level details, respectively, for the unseen RT target samples. With this *CoGAN-R*, we not only transfer the semantics from the source domain to the target domain but also transfer the domain shift from the IRT to the RT.

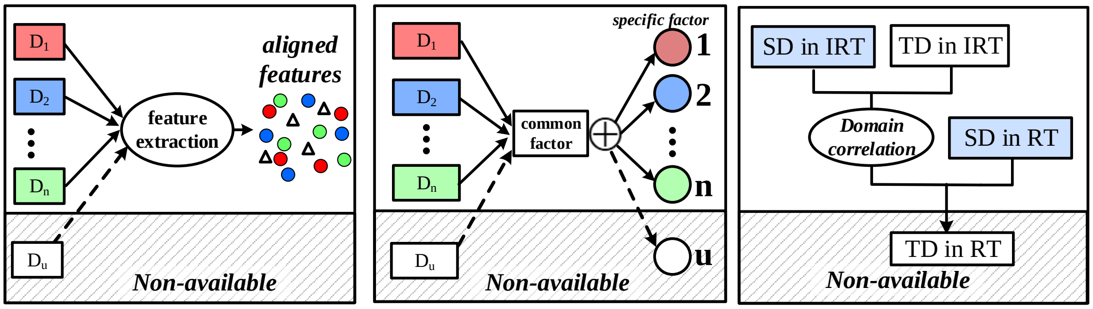
In summary, our contributions are threefold:

- (i) We propose a simple but effective strategy for preserving the domain shift across tasks by bridging the two CoGANs with a GAN. While the CoGANs capture the joint distributions of dual-domain samples, the GAN models the domain shift. In the absence of the target-domain data, we propose a new method for training the CoGAN for the RT under the supervision of both the source-domain data and the domain shift.
- (ii) We address the ZSDA problem without relying on the correspondence between the data samples across two domains in the IRT. Thus, our method has a broader range of applications than that reported by Peng *et al.* [15], which is applicable only when the correspondences are available. In addition, by modeling the domain shift explicitly with a GAN, our method can learn it from a single IRT and transfer it to multiple RTs.
- (iii) In the semantic segmentation task, we propose a method for learning a depth-based model from RGB images via domain shift transfer from the synthetic data to the real data. In our method, we train the network using a greedy approach (from individual instances to the whole scenes) and feed the CoGAN both the semantic label map and the instance-level boundary map to facilitate training.

II. Related Work

Zero-shot domain adaptation (ZSDA) [15] or domain generalization (DG) [22]–[24] refers to a group of domain adaptation tasks in which the target-domain data are not available in the training stage. For example, we may expect to learn a model for a camera even if the conditions of its working environment (such as the captured views and the lighting conditions) remain unknown. In this case, we can only access the source-domain samples, which could be high-quality images that were captured in an ideal environment. In contrast, the target domain that is determined by the working environment is not accessible. Fig. 2 summarizes the three strategies that have been primarily utilized by researchers to solve the ZSDA problem.

The first strategy learns domain-invariant features, which are applicable in both the source domain and the target domain. Muandet *et al.* [23] proposed domain-invariant component analysis for minimizing the dissimilarity across domains. Ghifary *et al.* [24] proposed a multi-task auto-encoder that can transform data samples from one domain to the other and, hence, learn features that are robust against the variation of domain labels. Ilse *et al.* [25] introduced a domain invariant autoencoder that learns three types of independent latent representations. Carlucci *et al.* [26] adopted a jigsaw puzzle as the self-supervised task for representation learning. Li *et al.* [27] trained a network in the episodic framework by decomposing it into one feature extraction module and one classifier module. Li *et al.* [28] extended adversarial autoencoders to align domains and matched the distribution of representations to a predefined distribution. Li *et al.* [35] proposed a conditional invariant adversarial network for learning



(a) Feature alignment (b) Common factor learning (c) Irrelevant task assistance

Fig. 2: Three strategies for addressing ZSDA tasks. (a) The first strategy learns a feature extraction procedure for producing aligned features from multiple source domains ($\{D_1, D_2, \cdots, D_n\}$) and applies the same procedure on the non-available target domain (D_u). (b) The second strategy decomposes each domain into a common factor and a domain-specific factor and applies the common factor to the target domain. (c) The third strategy solves the ZSDA task with the assistance of an IRT, where both the source domain (SD) and the target domain (TD) data are available. Typically, the domain correlation is learned from the IRT and transferred to the RT.

domain-invariant representations and aligning the conditional distributions across domains. Ding and Fu [36] introduced low-rank constraints for aligning multiple domain-specific neural networks. Qiao *et al.* [22] minimized the Wasserstein distance between the available source domain and various augmented domains.

The second strategy identifies a common factor from multiple source domains and applies it to the unseen target domain. Some methods [29], [30] assume that each domain is determined jointly by a globally common factor and a domain-specific factor. While the common factor generalizes well for the unseen domains, the domain-specific factor can be regarded as a bias for the associating domain. Khosla *et al.* [29] exploited the visual world by eliminating the bias from various training datasets and expected the learned model to perform well on unseen datasets. Later, Li *et al.* [30] utilized a similar strategy to design neural networks. Yang and Hospedales [37] applied manifold-valued data regression to estimate the unseen data and their labels. Kodirov *et al.* [2] formulated the zero-shot learning problem as a ZSDA task and correlated the source domain with the target domain via a shared semantic embedding space. To conduct ZSDA without relying on any semantic descriptors, Kumagai and Iwata [38] proposed latent domain vectors for representation learning and knowledge transfer across domains. Instead of designing a model, Li et al. [17] proposed a model agnostic training procedure that utilizes meta-learning approaches.

The third strategy realizes ZSDA with the assistance of an irrelevant task (IRT) for which the data from both domains are available. This strategy learns the correlation between two domains based on the available dual-domain data in the IRT and transfers the correlation to the task of interest. In contrast to the first two strategies, which rely on the availability of multiple source domains, this strategy involves only a single source domain. ZDDA [15] learns domain-invariant representations by minimizing the distance between the *paired samples*. In this way, the target classifier is trainable based on the source representations, namely, the approximations of the target representations. However, the application scope of ZDDA [15] is weakened by its reliance on a large set of *paired samples* in IRT. To learn with independent source samples and target samples in IRT, CoCoGAN [16] adopts a conditional structure to realize global alignment across tasks in both domains. However, it does not consider the category-level alignment between domains, and one category in RT may align with two different categories in IRT across the two domains. As a result, CoCoGAN cannot guarantee that the two tasks have a shared domain shift or that the target representations in RT are discriminant. Our method overcomes these two problems by explicitly defining the domain shift to be the distribution of the representation difference between the *paired samples* and transferring the domain shift across tasks. In comparison with both ZDDA [15] and CoCoGAN [16], the proposed method has the advantage that the domain shift is directly transferable from one task to multiple tasks whenever the same pair of domains are involved. A more detailed analysis and comparisons in terms of the learning logic between our proposed method and the state-of-the-art approaches are presented in Section IV-D.

III. Background

A. Generative Adversarial Networks (GANs)

A GAN [34] sets up a game between two competing networks: a generator (g) and a discriminator (f). The generator transforms a random vector $z \sim p_z$ into an image $g(z)$ that is expected to be indistinguishable from the real image $x \sim p_x$. The discriminator outputs a single scalar $f(t) \in [0, 1]$ and uses it to estimate the probability that its input t is drawn from p_x. These two competing networks are trained jointly in a minmax game by optimizing the following objective function:

$$\max_g \min_f V(f, g) \equiv E_{x\sim p_x}[-\log f(x)] + E_{z\sim p_z}[-\log(1 - f(g(z)))], \quad (1)$$

where E denotes the expected value. While the discriminator f minimizes $V(f, g)$ to distinguish the fake images from the real images, the generator g maximizes $V(f, g)$ to generate images that approximate the real images with increasing accuracy. It is shown that Eq. (1) measures the Jensen-Shannon divergence between the distribution of the real data and that of the generated data [34]. The distribution of the generated image $g(z)$ converges to p_x, and the discriminator always produces 0.5; *i.e.,* the generated images are indistinguishable from the real images.

B. Coupled Generative Adversarial Networks (CoGANs)

While a GAN captures the distribution of the data samples from a single domain, a CoGAN aims at capturing the joint distribution of the images from two different domains [33]. Let $x_1 \sim p_{x_1}$ and $x_2 \sim p_{x_2}$ be images from two different domains, where p_{x_1} and p_{x_2} are the marginal distributions. The CoGAN [33] learns the joint distribution of the data samples (p_{x_1,x_2}) based on samples that are drawn individually from the two marginal distributions (*i.e.,* p_{x_1} and p_{x_2}). The CoGAN consists of a pair of GANs, namely, GAN_1 and GAN_2, each of which corresponds to a domain. Let g_i be the generator and f_i be the discriminator in $\text{GAN}_i (i = 1, 2)$. The generators synthesize the *paired samples* $(g_1(z), g_2(z))$, which are indistinguishable from the real samples, based on the shared random vector z. The images $(g_1(z), g_2(z))$ are expected to have a correspondence instead of merely the same category label. Generally, two *paired samples* represent the same entity, such as a color image and a depth image of the same scene.

The CoGAN enforces a weight-sharing constraint in the layers for high-level concept processing. Mathematically, it solves the following optimization problem:

$$
\begin{aligned}
&\max_{g_1,g_2} \min_{f_1,f_2} V(f_1, f_2, g_1, g_2) \equiv \\
&\qquad E_{x_1 \sim p_{x_1}}[-\log f_1(x_1)] + E_{z \sim p_z}[-\log(1 - f_1(g_1(z)))] \\
&\qquad +E_{x_2 \sim p_{x_2}}[-\log f_2(x_2)] + E_{z \sim p_z}[-\log(1 - f_2(g_2(z)))], \\
&s.t. \quad \theta_{g_1^j} = \theta_{g_2^j}, \qquad\qquad 1 \leq j \leq n_g \\
&\qquad \theta_{f_1^{n_1-k}} = \theta_{f_2^{n_2-k}}, \quad 0 \leq k < n_f.
\end{aligned} \tag{2}
$$

Here, we use $\{\theta_{g_i^j} | 1 \leq j \leq n_g\}$ to denote the shared parameters of the $n_g \in \mathbb{Z}_{\geq 0}$ top layers in the two generators and $\theta_{f_i^{n_i-k}} (0 \leq k \leq n_f - 1)$ to denote the shared parameters of the $n_f \in \mathbb{Z}_{\geq 0}$ bottom layers in the two discriminators. Let $n_i (i = 1, 2)$ denote the number of layers in the discriminator f_i. These two constraints force the generators (or discriminators) to decode (or encode) the high-level semantics in the same way so that the CoGAN can learn a joint distribution of dual-domain images without any tuple of corresponding images.

IV. Proposed Method

A. Problem Definition

Following [39], we define a domain as $D = \{X, P(X)\}$, where X denotes the sample space and $P(X)$ denotes its marginal probability distribution. A task $T = \{Y, P(Y|X)\}$ is composed of a label space Y and its conditional probability distribution $P(Y|X)$ given the data samples X. We consider two tasks $T_1 = \{Y_1, P(Y_1|X_1)\}$ and $T_2 = \{Y_2, P(Y_2|X_2)\}$ to be the same if they share the label set. In other words, $T_1 = T_2$ if and only if $Y_1 = Y_2$, whereas X_1 and X_2 may be samples from different domains.

In RT, the label space is Y^r, the source domain is $D_s^r = \{X_s^r, P(X_s^r)\}$, and the target domain is $D_t^r = \{X_t^r, P(X_t^r)\}$. As a result, the RT can be described as $T^r = \{Y^r, P_s^r(Y^r|X_s^r)\} \cup \{Y^r, P_t^r(Y^r|X_t^r)\}$. Due to the domain shift, $P(X_s^r) \neq P(X_t^r)$ and $P_s^r(Y^r|X_s^r) \neq P_t^r(Y^r|X_t^r)$. Similarly, the IRT is $T^{ir} = \{Y^{ir}, P_s^{ir}(Y^{ir}|X_s^{ir})\} \cup \{Y^{ir}, P_t^{ir}(Y^{ir}|X_t^{ir})\}$, with the label space Y^{ir}, the source domain $D_s^{ir} = \{X_s^{ir}, P(X_s^{ir})\}$, and the target domain $D_t^{ir} = \{X_t^{ir}, P(X_t^{ir})\}$. As both RT and IRT involve the same pair of domains, the sample sets X_s^r and X_s^{ir} (X_t^r and X_t^{ir}) are different subsets of the same sample space. As shown in Fig. 1, while both X_s^r (*i.e.,* MNIST) and X_s^{ir} (*i.e.,* Fashion-MNIST) are gray images, both X_t^r (*i.e.,* MNIST-M) and X_t^{ir} (*i.e.,* Fashion-MNIST-M) are color images.

In principle, our ZSDA aims at learning the conditional probability distribution $P_t^r(Y^r|X_t^r)$ from the labeled source-domain data (*i.e.,* X_s^r) in RT and the labeled dual-domain samples (*i.e.,* X_s^{ir} and X_s^{ir}) in IRT. This is a challenging problem since neither $x_t^r \in X_t^r$ nor its label $y_t^r \in Y^r$ is available for training.

B. Main Idea

In both RT and IRT of our ZSDA problem, a sample in the source domain has a correspondence in the target domain. For convenience of description, we refer to two corresponding samples as *paired samples*. Typically, *paired samples* are originate from the same object, examples of which include an image and its edge version, and an RGB image and a depth image of the same scene. With the network structure of CoGAN [33], our method does not rely on the correspondences between the dual-domain samples in the training stage.

We define the domain shift as the distribution of the element-wise representation difference between the *paired samples*. Given a pair of domains, we assume that the domain shift is task-independent, namely, that the element-wise representation difference between the *paired samples* follows the same distribution across tasks. This enables us to learn the domain shift based on the available dual-domain samples in IRT and transfer it to RT. With the available source-domain data in RT and the transferable domain shift that is learned from IRT, we can synthesize the target-domain data of the RT and solve the ZSDA problem. To realize the above, two key issues are identified: (i) the learning and the explicit modeling of the domain shift based on the non-corresponding dual-domain samples in IRT; and (ii) the transfer of the domain shift from IRT to RT for target-domain data generation in RT.

To resolve the first issue, we learn the domain shift from a CoGAN and model the domain shift with a GAN. First, we train a *CoGAN-IR* for the IRT to capture the joint distribution of the available dual-domain samples. The *CoGAN-IR* correlates the data samples from two domains with their common high-level semantics. We use this *CoGAN-IR* to generate a set of *paired samples* and extract their representations. Then, we train a *GAN*^{shift} to capture the distribution of the element-wise difference between the representations of these *paired samples* and characterize the domain shift with this distribution.

To resolve the second issue, we train a *CoGAN-R* for the RT and enforce it to carry the same domain shift as the *CoGAN-IR* does for the IRT. In other words, we expect the element-wise difference between representations (produced by the CoGANs) of the *paired samples* to follow the same distribution in two tasks. To realize this objective, we propose a training method for CoGAN based on a strategy of domain shift preservation across tasks.

C. Network Training and Testing

Let $x_s^{ir} \in X_s^{ir}$ ($x_s^r \in X_s^r$) and $x_t^{ir} \in X_t^{ir}$ ($x_t^r \in X_t^r$) be the samples in IRT (RT). In our ZSDA problem, x_t^r is not available for training. For simplicity, we use $R(x_s^{ir})$, $R(x_s^r)$, $R(x_t^{ir})$, and $R(x_t^r)$ to denote the representations of these samples, although the representation extraction procedure $R(\cdot)$ varies with the domain and the task.

Fig. 3 illustrates the network structure of our model, which consists of *CoGAN-IR*, *GAN*^{shift}, and *CoGAN-IR*. We use *CoGAN-IR* to not only capture the joint distribution of (x_s^{ir}, x_t^{ir}) in IRT but also lay a data foundation for domain shift modeling by generating the *paired samples*. Based on the observation that the domain shift introduces the difference between the *paired samples*, we model the domain shift with a *GAN*^{shift} that captures the distribution of element-wise difference $\delta_x^{ir} = R(x_s^{ir}) \ominus R(x_t^{ir})$ between the *paired samples* (x_s^{ir}, x_t^{ir}) in the representation space. In this way, we encode the domain shift with a network structure and, hence, render it more convenient to transfer. We transfer the domain shift across tasks by enabling *CoGAN-R*, which captures the joint distribution of dual-domain samples of RT, to carry the domain shift that is encoded in *GAN*^{shift}. With this *CoGAN-R*, we not only transfer the semantics from the source domain to the target domain but also transfer the domain shift from IRT to RT. To realize effective ZSDA, we propose a three-stage training procedure and a four-stage testing procedure, both of which are described in detail as pseudo-codes in Algorithm 1.

1) Training: The first stage trains *CoGAN-IR* to learn the joint distribution of the *paired samples* in IRT by optimizing the following objective function:

$$\begin{aligned}\max_{g_1^{ir}, g_2^{ir}} \min_{f_1^{ir}, f_2^{ir}} & V(f_1^{ir}, f_2^{ir}, g_1^{ir}, g_2^{ir}) \equiv \\ & E_{x_s^{ir} \sim p_{x_s^{ir}}}[-\log f_1^{ir}(x_s^{ir})] + E_{z^{ir} \sim p_{z^{ir}}}[-\log(1 - f_1^{ir}(g_1^{ir}(z^{ir})))] \\ & + E_{x_t^{ir} \sim p_{x_t^{ir}}}[-\log f_2^{ir}(x_t^{ir})] + E_{z^{ir} \sim p_{z^{ir}}}[-\log(1 - f_2^{ir}(g_2^{ir}(z^{ir})))],\end{aligned} \tag{3}$$

where g_1^{ir} and g_2^{ir} are generators and f_1^{ir} and f_2^{ir} are discriminators. As CoGAN can learn joint distribution from marginal distributions, this stage does not rely on the correspondences between samples in two domains. Instead, the first training stage uses only the samples that are individually drawn from the source domain and those from the target domain. This *CoGAN-IR* can generate a set of *paired samples* in IRT, which have correspondences and share the same high-level concepts. In other words, *CoGAN-IR* encodes the correlations between the two domains.

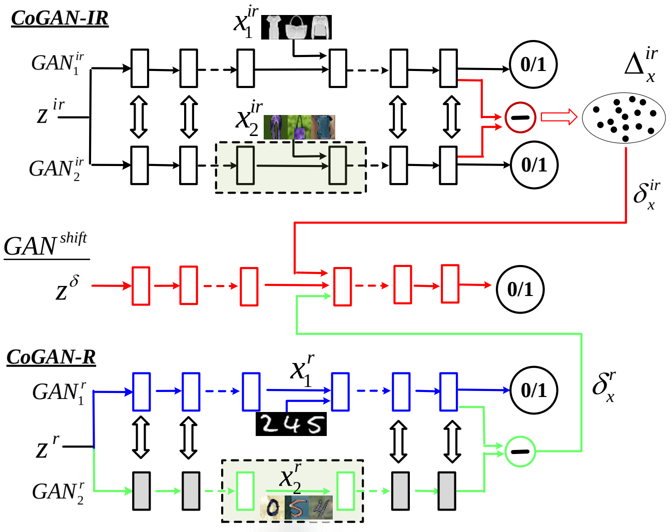


Fig. 3: Illustration of our proposed model structure, where *CoGAN-IR* models the joint distribution of (x_s^{ir}, x_t^{ir}) in IRT and *CoGAN-R* models the joint distribution of (x_s^r, x_t^r) in RT. *GAN*^{shift} learns the domain shift by capturing the distribution of the element-wise representation difference $\delta_x^{ir} = R(x_s^{ir}) \ominus R(x_t^{ir})$ between the *paired samples* (x_s^{ir}, x_t^{ir}). The domain shift is transferred from IRT to RT by restricting *CoGAN-R* to carry the domain shift that is learned by *GAN*^{shift}, namely, $p_{\delta_x^r} = p_{\delta_x^{ir}}$. The shared layers in each CoGAN are marked by vertical double-headed arrows.

The second stage trains *GAN*^{shift} to model the domain shift by capturing the distribution of the element-wise difference between the *paired samples* in the representation space. With *CoGAN-IR*, we can easily generate a set of *paired samples* $(\tilde{x}_s^{ir}, \tilde{x}_t^{ir})$ for IRT and obtain their element-wise representation difference as $\delta_x^{ir} = R(\tilde{x}_s^{ir}) \ominus R(\tilde{x}_t^{ir})$. As the *paired samples* have a correspondence, their representation difference is primarily caused by the domain shift. Therefore, we use *GAN*^{shift} to capture the distribution of the representation difference δ_x^{ir} and train it by optimizing the following objective function:

$$\begin{aligned}\max_{g^{\delta}} \min_{f^{\delta}} V(f^{\delta}, g^{\delta}) \equiv & E_{\delta_x^{ir} \sim p_{\delta_x^{ir}}}[-\log f(\delta_x^{ir})] \\ & + E_{z^{\delta} \sim p_{z^{\delta}}}[-\log(1 - f^{\delta}(g^{\delta}(z^{\delta})))],\end{aligned} \tag{4}$$

where z^{δ} is a random variable, g^{δ} is the generator, and f^{δ} is the discriminator.

The third stage trains the *CoGAN-R* to learn the joint distribution of the *paired samples* in RT. The *CoGAN-R* consists of a pair of GANs: *GAN*$_1^r = \{g_1^r, f_1^r\}$ for the source domain and *GAN*$_2^r = \{g_2^r, f_2^r\}$ for the target domain. Due to the non-availability of X_t^r, we cannot train the *CoGAN-R* via the classic method [33]. In this work, we exploit two supervisory principles to train the two branches of the *CoGAN-R* individually as follows:

- The source-domain branch *GAN*$_1^r$ captures the distribution of the available X_s^r;
- Both the *CoGAN-IR* and the *CoGAN-R* carry the same domain shift, *i.e.*, $p_{\delta_x^{ir}} = p_{\delta_x^r}$.

Algorithm 1 Proposed method

```
Training Procedure
  Stage 1 Train CoGAN-IR to capture the joint distribution of $(x_s^{ir}, x_t^{ir})$ by optimizing Eq. (3).
  Stage 2 Model the domain shift explicitly:
    2.1 Generate a set of paired samples $(\tilde{x}_s^{ir}, \tilde{x}_t^{ir})$ with CoGAN-IR;
    2.2 Extract the representations $R(\tilde{x}_s^{ir})$ and $R(\tilde{x}_t^{ir})$ with CoGAN-IR and calculate their element-wise difference as $\delta_x^{ir} = R(\tilde{x}_s^{ir}) \ominus R(\tilde{x}_t^{ir})$;
    2.3 Train $GAN^{shift}$ to capture the distribution of $\delta_x^{ir}$ by optimizing (4).
  Stage 3. Train CoGAN-R to capture the joint distribution of $(x_s^r, x_t^r)$:
    3.1 Train $GAN_1^r$ to capture the distribution of $x_s^r$;
    3.2 Initialize the parameters of the non-shared layers in $GAN_2^r$ with their counterparts in $GAN_2^{ir}$;
    3.3 Fix the shared layers in $GAN_2^r$ and train the non-shared layers by optimizing Eq. (6).
Testing Procedure
  Stage 1 Train the classifier $h_s(\cdot)$ for $X_s^r$ in the source domain;
  Stage 2 Generate a set of paired samples $(\tilde{x}_s^r, \tilde{x}_t^r)$ with CoGAN-R;
  Stage 3 Predict the label of $\tilde{x}_s^r$ and assign it to $\tilde{x}_t^r$ for each sample pair $(\tilde{x}_s^r, \tilde{x}_t^r)$;
  Stage 4 Train a classifier $h_t(\cdot)$ for $\tilde{x}_t^r$ with the constraint $h_s(\tilde{x}_s^r) = h_t(\tilde{x}_t^r)$.
```

Correspondingly, first, we consider the source-domain branch, *i.e.,* GAN_1^r, as an independent GAN and train it to well capture the distribution of the available X_s^r. Following that, we fix the parameters of GAN_1^r and adapt the parameters of GAN_2^r to preserve the domain shift by treating its shared layers and non-shared layers differently. The target-domain branch GAN_2^r is an approximation of an independent GAN, *i.e.,* $GAN_2^{r+} = \{g_2^{r+}, f_2^{r+}\}$, which optimizes the following objective function:

$$\max_{g_2^{r+}} \min_{f_2^{r+}} V(f_2^{r+}, g_2^{r+}) \equiv E_{x_t^r \sim p_{x_t^r}}[-\log f_2^{r+}(x_t^r)] + E_{z^{r+} \sim p_{z^{r+}}}[-\log(1 - f_2^{r+}(g_2^{r+}(z^{r+})))]. \tag{5}$$

However, as X_t^r is not available, Eq. (5) is not directly solvable.

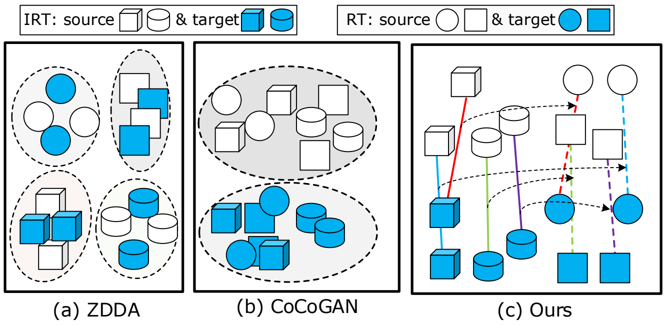
The shared layers are the layers that deal with the high-level semantic concepts for both GAN_1^r and GAN_2^r. They are the bottom layers in the generators and the top layers in the discriminators. We mark the shared layers by vertically double-headed arrows in Fig. 3. The parameters of these shared layers in GAN_2^r are simply copied from GAN_1^r to guarantee that the two branches (*i.e.,* GAN_1^r and GAN_2^r) deal with the high-level semantics in the same way.

The non-shared layers in GAN_2^r deal with the low-level features in the target domain. As both the samples x_2^{ir} and x_2^r originate from the same domain, we expect to process their low-level features similarly. Thus, we initialize the parameters of the non-shared layers in GAN_2^r by the parameter values that were learned in GAN_2^{ir} (marked by dotted rectangles in Fig. 3). These initialized parameters are further fine-tuned for domain shift preservation across tasks. We adapt the parameters of the non-shared layers and let *CoGAN-R* to produce $\delta_x^r = R(\tilde{x}_s^r) \ominus R(\tilde{x}_t^r)$, which is indistinguishable from δ_x^{ir}, where $\tilde{x}_s^r$ and $\tilde{x}_t^r$ are the *paired samples* that are generated from the same noise z^r. Mathematically, we fine-tune the parameters of the non-shared layers based on the following objective function:

$$\min_{g_2^r, f_2^r} V(g_2^r, f_2^r) \equiv E_{z^r \sim p_{z^r}}[-\log(1 - f^\delta(\delta_x^r)]. \tag{6}$$

In summary, while the shared layers in GAN_2^r are initialized based on GAN_1^r and frozen for high-level concept consistency, the non-shared layers in GAN_2^r are initialized based on GAN_2^{ir} to mimic its low-level feature processing. The non-shared layers of GAN_2^r are further fine-tuned for domain shift preservation across tasks.

2) Testing: Considering the classification task as an example, we evaluate our method and learn a model for the unseen target domain in RT with the following four stages. The first stage trains a classifier $h_s(\cdot)$ for the source domain with the available data X_s^r. The second stage generates a set of *paired samples* $(\tilde{x}_s^r, \tilde{x}_t^r)$ with *CoGAN-R*. The third stage predicts the label of $\tilde{x}_s^r$ by $h_s(\tilde{x}_s^r)$ and assigns it to $\tilde{x}_t^r$. The fourth stage trains a classifier $h_t(\cdot)$ for the generated target-domain samples $\tilde{x}_t^r$ with the constraint $h_s(\tilde{x}_s^r) = h_t(\tilde{x}_t^r)$, which ensures that the *paired samples* have the same high-level concepts.



(a) ZDDA (b) CoCoGAN (c) Ours

Fig. 4: Ideal resulting representation spaces of three methods (best viewed in color). (a) ZDDA: representation alignment across domains in both tasks; (b) CoCoGAN: representation alignment across tasks in both domains; (c) Ours: alignment of element-wise differences (marked by solid lines for RT and dotted lines for IRT) across tasks.

D. Discussion

To highlight the differences between our proposed model and the state-of-the-art models, we conduct a theoretical analysis in this section to compare our method with two established methods, namely, ZDDA [15] and CoCoGAN [16]. Fig. 4 visualizes all the ideal result representation spaces for these three compared methods, which clearly show that the three compared methods address the problem of ZSDA with different alignment schemes. While ZDDA [15] realizes representation alignment across domains in both tasks and CoCoGAN [16] realizes representation alignment across tasks in both domains, our method learns to align the cross-domain difference (marked by solid lines for RT and dotted lines for IRT in Fig. 4 (c)) in the two tasks.

In principle, ZDDA [15] learns to approximate the target representations by their correspondences in the source domain. It achieves the approximation in IRT by minimizing the representation difference between the *paired samples*. This approximation is expected to hold in RT, too. The underlying logic behind this method can be expressed as follows:

$$\min_{(x_s^{ir},x_t^{ir})} \|R(x_s^{ir})-R(x_t^{ir})\|^2 \Rightarrow R(x_s^r)\approx R(x_t^r), \tag{7}$$

where (x_s^{ir},x_t^{ir}) and (x_s^r,x_t^r) are *paired samples* in two tasks. Fig. 4 (a) visualizes the ideal representation space of ZDDA, in which every target sample is close to its corresponding source sample for both tasks.

In a conditional network, CoCoGAN [16] establishes a cross-task alignment in the source domain by maximizing the overlap between the representations of two tasks (*i.e.,* $R(X_s^{ir})$ and $R(X_s^r)$), and this alignment is expected to hold in the target domain, too. Its learning logic can be described as follows:

$$\max_{X_s^{ir},X_s^r} \big(R(X_s^{ir})\cap R(X_s^r)\big) \Rightarrow R(x_t^r)\in R(X_t^{ir}), \tag{8}$$

where $R(x_t^r)\in R(X_t^{ir})$ denotes that the target representation $R(x_t^r)$ in RT lies in the region that is spanned by the IRT representations, *i.e.,* $R(X_t^{ir})$. Compared with ZDDA [15], which learns to realize cross-domain approximation based on corresponding samples in IRT, CoCoGAN learns based on independent samples. While CoCoGAN can align RT and IRT globally in both domains, it does not consider the category-level alignment and, thus, cannot ensure that the two tasks have a shared domain shift. As a result, CoCoGAN cannot guarantee that the target representations are discriminative in RT; an example is presented in Fig. 4 (b). The *square* and the *circle* (two categories in RT) are indistinguishable from each other in the target domain, which is due to the asymmetric category-level alignments in two domains. The *square* is aligned with different categories in two domains, namely, a *cylinder* in the source domain and a *cube* in the target domain.

In contrast, our proposed method enforces a shared domain shift between the tasks, and the logic can be expressed as follows:

$$GAN^{shift}\sim\delta_x^{ir}\Rightarrow\delta_x^r\sim GAN^{shift}, \tag{9}$$

where $\delta_x^{ir}=R(\tilde{x}_s^{ir})\ominus R(\tilde{x}_t^{ir})$ and $\delta_x^r=R(\tilde{x}_s^r)\ominus R(\tilde{x}_t^r)$ are the element-wise differences between *paired samples* in the representation space. In the resulting representation space, as shown in Fig 4 (c), the differences between the *paired samples* in RT (marked by dotted lines) are indistinguishable from those in IRT (marked by solid lines). While both ZDDA and CoCoGAN consider the relationship between representations across tasks and domains, our method directly considers the difference between the representations (*i.e.,* domain shift), thereby providing a better implementation of the original approach of borrowing the domain shift from IRT.

V. Experiments

A. Image Classification

1) Dataset: First, we conduct experiments on four datasets of gray-scale images. Both MNIST (D_M) [31] and Fashion-MNIST (D_F) have $70K$ samples in 10 classes, which consist of handwritten digit images and fashion images, respectively. NIST (D_N) [40] and its extension EMNIST (D_E) [41] include letter images. In D_N, we use both the lowercase letter images and the uppercase letter images; thus, we consider $41K$ images from 52 classes. In D_E, we merge the uppercase and lowercase letters to form a balanced 26-class dataset with $14K$ images.

These datasets are in the gray domain (G–dom). We create three more domains from G–dom for adaptation. As reported by Ganin *et al.* [1], a sample $I_c=|I-P|$ in the color domain (C–dom) combines the grayscale image $I\in R^{m\times n}$ with a patch $P\in R^{m\times n\times 3}$ in each channel, where the patch P is randomly cropped from a color image in BSDS500 [42]. For an image I, the canny detector transforms it to its edge image I_e in the edge domain (E–dom), and the operation $I_n=255-I$ derives the negative image I_n in the negative domain (N–dom). Fig. 5 presents 48 example images from four domains.

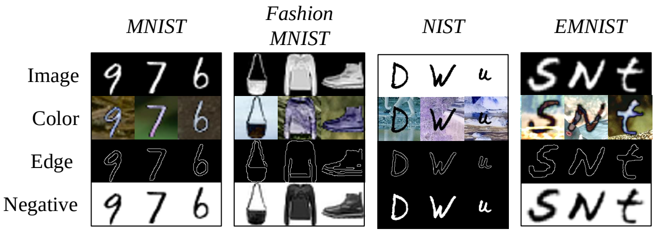


Fig. 5: Example images from 4 datasets and their counterparts in 3 domains.

Second, we evaluate our method on two publicly available datasets. **Office-Home** [43] consists of more than $15K$ images from 65 categories. The images correspond to four domains, namely, Art (**Ar**), Clipart (**Cl**), Product (**Pr**), and Real-world (**Rw**). In total, we have 12 combinations of *(source, target)* domain pairs. **VisDA2017** [44] contains approximately $280K$ images from 12 categories. This dataset explores the relationship between the synthetic images and the real images. The synthetic images are generated by rendering 3D models under various lighting conditions and angles.

2) Implementation details: We use convolutional neural networks to implement both GAN and CoGAN. *CoGAN-IR* and *CoGAN-R* have the same network structure so that the domain shift is transferable between them, and the two branches inside each CoGAN also share the same structure. For the first four datasets, the generators have seven transposed convolutional layers with stride 2 for up-sampling, and the discriminators have five convolutional layers with stride 2 for down-sampling and two additional convolutional layers for binary classification. For the last two challenging datasets, we use DCGAN [45] as the backbone to construct the four branches of the two CoGANs. We transform the output of the last convolutional layer of the discriminator into a column

TABLE II: The accuracy (%) on D_M, D_F, D_N and D_E (**bold-red** indicates the best and **bold-black** indicates the 2nd best).

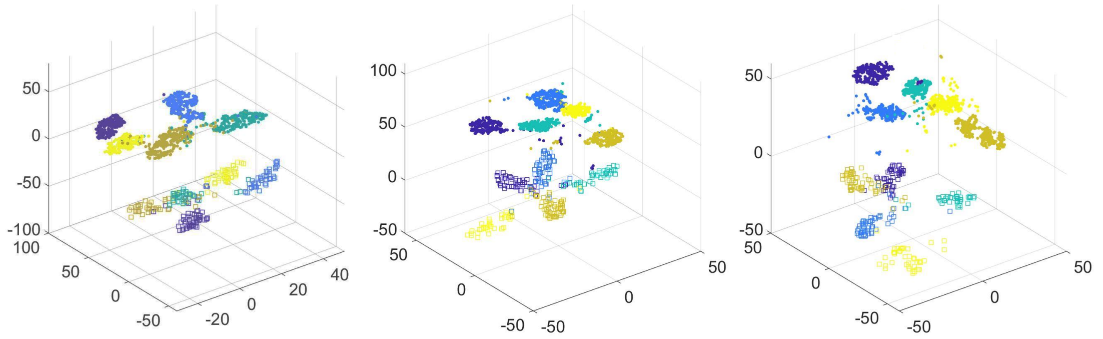
(Source, Target)	RT	MNIST (D_M)			Fashion-MNIST (D_F)			NIST (D_N)		EMNIST (D_E)	
	IRT	D_F	D_N	D_E	D_M	D_N	D_E	D_M	D_F	D_M	D_F
$(G\text{–}dom, C\text{–}dom)$	DIVA [25]	75.9±2.8	86.3±3.0	87.1±1.4	52.5±5.5	**57.3±1.1**	62.4±4.6	**44.1±3.0**	39.5±4.8	72.4±3.9	50.7±1.1
	JiGen [26]	61.6±2.5	78.4±4.1	73.1±3.4	47.7±2.6	49.6±4.5	60.9±2.1	39.2±4.1	37.0±1.7	58.9±4.9	49.9±3.5
	MATE [24]	**78.6±2.9**	82.7±6.5	89.7±2.4	50.5±3.4	54.6±5.7	62.1±5.2	43.5±5.8	44.1±2.2	**75.7±5.4**	51.4±7.0
	Epi-FCR [27]	74.3±5.5	86.6±3.5	88.8±3.5	**57.4±1.2**	56.8±5.4	59.9±4.7	38.6±2.0	43.6±3.7	71.7±4.3	54.1±4.3
	MMD-AAE [28]	58.8±4.1	69.3±1.9	69.6±5.9	39.3±3.0	42.5±5.2	49.8±3.7	36.2±5.8	40.0±3.6	63.0±3.6	44.0±5.4
	CoCoGAN [16]	78.1±0.6	**92.4±1.0**	**95.6±1.2**	56.8±1.3	56.7±0.9	**66.8±1.5**	41.0±1.1	**44.9±1.7**	**75.0±0.9**	54.8±0.9
	ZDDA+PD [15]	73.4±1.9	89.8±2.4	93.5±1.5	50.7±0.9	43.9±1.1	**64.7±1.6**	34.8±2.8	40.3±2.8	69.2±3.0	**56.7±3.7**
	ZDDA+DC [15]	69.5±1.4	83.9±2.3	86.3±2.7	45.7±2.8	41.2±2.0	59.6±2.1	29.3±1.5	37.4±1.7	52.7±2.6	40.8±4.1
	Ours	**82.9±1.3**	**94.4±2.8**	**95.3±1.4**	**59.1±0.8**	**58.6±2.2**	63.1±2.7	**46.4±2.9**	**46.5±4.0**	73.6±2.0	**58.3±1.6**
$(G\text{–}dom, E\text{–}dom)$	DIVA [25]	**81.7±3.1**	87.2±7.1	89.5±3.3	61.3±3.9	**58.4±3.6**	67.5±4.3	44.0±2.5	41.3±3.7	75.8±2.9	54.7±6.8
	JiGen [26]	68.7±2.4	78.0±5.5	72.4±7.2	59.4±4.9	46.0±5.8	59.1±2.1	42.8±2.4	32.6±7.4	64.2±3.6	54.6±3.8
	MATE [24]	76.8±2.3	89.9±6.1	88.6±4.3	**62.4±4.0**	56.3±1.5	67.6±2.6	47.3±4.2	34.4±2.6	69.4±7.1	56.4±4.1
	Epi-FCR [27]	74.7±7.3	91.9±2.5	87.6±5.0	58.7±5.8	56.5±3.8	65.6±4.0	**48.3±3.5**	**45.5±3.9**	73.2±4.4	57.7±4.3
	MMD-AAE [28]	57.7±3.3	72.5±2.3	74.7±3.8	52.7±4.7	50.2±4.3	59.4±3.7	43.5±3.2	36.6±3.3	65.7±4.1	47.3±5.8
	CoCoGAN [16]	79.6±1.2	**94.9±0.5**	**95.4±0.7**	61.5±1.1	57.5±1.4	**71.0±0.8**	48.0±0.8	36.3±1.2	**77.9±1.3**	58.6±1.4
	ZDDA+PD [15]	72.5±4.8	91.5±1.3	93.2±2.7	54.1±4.3	54.0±5.9	65.8±4.9	42.3±1.8	28.4±3.9	73.6±1.9	**60.7±3.9**
	ZDDA+DC [15]	68.4±1.6	84.7±2.0	86.4±1.3	47.7±3.2	51.2±1.9	57.7±1.3	41.2±3.0	31.2±2.1	66.4±2.7	53.2±1.6
	Ours	**84.7±2.4**	**93.9±1.6**	**95.5±2.1**	**64.8±1.0**	**60.4±2.3**	**73.3±1.8**	**50.2±2.0**	**43.4±1.4**	**79.2±1.4**	**64.7±2.6**
$(G\text{–}dom, N\text{–}dom)$	DIVA [25]	76.0±4.7	76.3±2.9	88.2±3.3	62.5±5.2	**55.3±4.7**	**70.6±4.0**	46.8±3.3	51.1±6.2	77.0±5.1	**63.4±2.3**
	JiGen [26]	62.5±4.1	72.9±3.1	79.4±5.2	54.7±3.8	47.7±4.7	58.9±5.4	36.5±7.9	41.5±4.7	67.9±4.4	55.1±3.1
	MATE [24]	76.1±4.8	81.5±5.1	85.1±3.2	**69.2±3.7**	53.2±5.5	66.3±3.9	**48.9±2.0**	54.5±6.4	75.9±6.6	59.8±5.3
	Epi-FCR [27]	77.8±5.8	84.3±1.1	86.1±3.8	68.0±3.0	54.8±3.8	67.7±5.7	45.3±7.9	**55.5±3.8**	78.7±5.6	61.0±3.8
	MMD-AAE [28]	**83.1±4.9**	43.7±4.9	64.4±7.1	46.2±5.5	20.8±2.6	59.2±7.1	43.8±5.2	45.7±8.5	69.4±0.9	48.2±5.4
	CoCoGAN [16]	80.3±1.1	87.5±0.7	**93.1±1.3**	66.0±1.1	52.2±0.8	69.3±0.6	45.7±0.5	53.8±1.0	**81.1±1.1**	56.5±1.1
	ZDDA+PD [15]	77.9±4.8	**88.2±2.8**	90.5±0.8	61.4±3.4	47.4±2.4	62.7±3.9	37.8±2.5	46.7±3.0	76.2±2.8	53.4±2.9
	ZDDA+DC [15]	67.2±0.8	76.3±2.9	84.2±1.8	56.3±1.6	47.7±2.4	56.1±3.7	32.6±2.5	30.6±1.6	62.1±1.5	43.4±3.5
	Ours	**83.6±1.7**	**89.1±1.6**	**94.2±2.4**	**71.1±1.3**	**58.0±2.3**	**70.3±1.9**	**49.6±3.0**	**57.4±1.6**	**82.1±2.5**	**65.2±2.2**
$(C\text{–}dom, G\text{–}dom)$	DIVA [25]	65.8±4.5	89.1±6.0	92.7±3.2	60.8±3.7	46.0±4.8	68.4±1.1	**50.5±4.4**	53.1±5.3	78.3±5.5	64.1±4.3
	JiGen [26]	56.9±6.5	72.7±4.0	80.0±2.5	50.2±2.3	32.2±3.1	51.0±3.9	37.1±4.1	49.6±2.5	68.8±3.0	52.4±6.2
	MATE [24]	69.9±5.8	**90.3±2.2**	86.5±3.8	58.6±3.3	**55.5±8.3**	**71.1±6.4**	45.1±4.4	**66.1±1.6**	64.7±3.0	**76.2±2.2**
	Epi-FCR [27]	70.8±6.1	86.2±3.8	90.8±4.9	**62.6±5.4**	52.8±4.5	66.3±1.7	48.7±2.2	55.8±3.8	**83.0±2.3**	66.6±6.5
	MMD-AAE [28]	55.0±6.8	73.9±2.2	75.1±4.3	40.0±7.1	41.2±5.6	63.9±0.8	38.2±4.3	38.1±6.7	62.2±5.4	63.8±4.7
	CoCoGAN [16]	**73.2±1.3**	89.6±0.6	**94.7±0.4**	61.1±0.9	50.7±1.2	70.2±1.1	47.5±1.1	57.7±1.8	80.2±1.7	67.4±1.4
	ZDDA+PD [15]	67.4±4.2	85.7±4.0	87.6±4.7	55.1±3.7	49.2±2.5	59.5±5.9	39.6±3.4	23.7±7.3	75.5±1.7	52.0±2.7
	ZDDA+DC [15]	56.7±1.6	72.0±2.0	76.4±2.3	48.4±2.1	45.2±1.9	53.7±1.8	32.8±1.7	20.6±2.1	69.2±2.6	47.6±1.6
	Ours	**76.2±1.8**	**94.6±0.9**	**95.2±0.6**	**63.2±1.2**	**54.3±1.0**	**73.6±0.8**	**53.8±1.4**	**61.4±1.1**	**84.3±1.2**	**71.5±1.0**

vector before feeding it into a sigmoid function. In both architectures, the last two layers in the generators and the first two layers in the discriminators are non-shared layers for low-level feature processing in different domains. In all datasets, the representation $R(x)$ of the input x is defined to be the feature map that is produced by the second-to-last layer in the discriminators of *CoGAN-IR*. We construct GAN^{shift} with five transposed convolutional layers in the generator and five convolutional layers in the discriminator. We vary the input noise z^{ir} to obtain $\Delta_x^{ir} = \{\delta_x^{ir} | \delta_x^{ir} = R(\tilde{x}_s^r(z^{ir})) \ominus R(\tilde{x}_t^r(z^{ir})), t = 1, 2 \cdots, N\}$ and use them to train GAN^{shift}.

For *CoGAN-IR*, we set the learning rate to 10^{-4}, the batch size to 32, and the weight decay to 0.005. We use the ADAM algorithm to update the parameters with $30K$ iterations. As reported in [33], the first momentum parameter is 0.5, and the second momentum parameter is 0.999. For GAN^{shift}, we initialize the parameters with a zero-centered normal distribution and optimize them via stochastic gradient descent (SGD). The batch size is 32, and the learning rate is 2×10^{-4}. GAN^{shift} is trained with $20K$ iterations. We train GAN_1^r in *CoGAN-R* with the same settings to train GAN_1^{ir}. After initializing the shared and non-shared layers of GAN_2^r separately (as detailed in IV-C), we finetune its parameters using SGD with $25K$ iterations. The learning rate is 10^{-4}, and the batch size is 16.

3) Results: To the best of our knowledge, only two previously established methods rely on an IRT for deep learning-based ZSDA from a single source domain, namely, ZDDA [15] and CoCoGAN [16]. We denote the ZDDA model that is trained with the paired data as the baseline ZDDA+PD. To provide a wider evaluation of our proposed method, we further introduce another baseline, namely, ZDDA+DC, by replacing the L_2-loss of ZDDA [15] with a binary domain classifier to render it applicable to datasets in which the *paired samples* are not available. We also compare our method with five established methods that learn from multiple ($\geq$ 2) source domains instead of relying on an IRT: DIVA [25], JiGen [26], MATE [24], Epi-FCR [27], and MMD-AAE [28]. As the training procedure of our method involves samples from three sources (X_s^{ir}, X_t^{ir}, and X_s^r), we also learn the baselines [24]–[28] based on samples from three sources for fair comparison. They are the three domains different from the target domain in *Office-Home* and $\{X_s^r, rot_{\alpha_1}(X_s^r), rot_{\alpha_2}(X_t^r)\}$ in the remaining five datasets, where $rot_\alpha(\cdot)$ (also adopted in [24], [25], [28]) denotes the operation of rotating the sample by an angle of α. In our experiments, we report the results with $\alpha_1 = \alpha_2 = 45°$, which can endow the source samples with sufficient diversity. The conclusions remain the same when the angle value is changed to 15°, 30°, or 60°.

With the first four datasets, we test the domain shift transfer across three classification tasks on four (source domain, target domain) pairs: (G, C), (G, E), (G, N), and (C, G). Given a pair of domains, there are ten possible pairs of (IRT, RT) in total. For increased similarity to real applications, our training data do not include the *paired samples*. We partition the dataset



(a) MNIST (b) Fashion-MNIST (c) NIST

Fig. 6: t-SNE visualization of the representations in two domains: "●" for *G–dom* and "□" for *C–dom* (best viewed in color).

of the IRT into two non-overlapping halves, and the training set consists of the source-domain samples in the first half and the target-domain samples in the second half.

Tab. II summarizes the average classification accuracy over ten runs. Our method performs the best on average and realizes the highest accuracy in 31 of 40 settings. As a representative method for extracting domain-invariant features from multiple domains, MATE [24] is the second best model and performs the best in four settings. According to the comparison between [24]–[28] and our method, our new strategy of domain shift transfer is promising for the task of ZSDA. In comparison with ZDDA+PD [15], our method realizes higher accuracy with less supervisory information (does not rely on the correspondences between dual-domain samples in IRT). In the adaptation from $G\text{–}dom$ to $C\text{–}dom$ with D_M as RT, our method realizes the accuracies of 94.4% with D_N as IRT and 95.3% with D_E as IRT. These accuracies exceed those of three established methods (including 86.7% in [46], 89.5% in [47], and 94.2% in [48]) that rely on the availability of the target-domain data in the training stage. With GAN^{shift} modeling the domain shift explicitly, our method performs efficiently in applications with many target domains as it can conduct shift transfer from one IRT to multiple RTs. In contrast, the previously established methods require a new round of training whenever either RT or IRT changes. Fig. 6 illustrates a visualization of representations in both $G\text{–}dom$ (marked by the solid dots) and $C\text{–}dom$ (marked by squares). For demonstration, we only visualize the first five categories in each of the three datasets. The discrepancy between these two domains is readily identifiable for each dataset, and the representations are discriminative.

TABLE III: Average accuracy (%) on *Office-Home*

Target	*Ar*			*Cl*			*Pr*			*Rw*		
Source	*Cl*	*Pr*	*Rw*	*Ar*	*Pr*	*Rw*	*Ar*	*Cl*	*Rw*	*Ar*	*Cl*	*Pr*
DIVA [25]	68.4			61.4			69.7			74.1		
JiGen [26]	64.0			58.5			76.5			**78.1**		
MATE [24]	65.7			50.5			74.6			65.0		
Epi-FCR [27]	72.1			**62.6**			**76.8**			72.8		
MMD-AAE [28]	71.5			57.3			65.7			68.5		
CoCoGAN [16]	66.7	57.6	69.2	62.2	53.4	51.3	69.5	74.0	65.8	74.5	66.4	71.7
ZDDA+DC [15]	67.4	60.9	68.1	53.2	40.6	43.4	61.4	57.0	50.3	68.8	68.4	62.4
Ours	72.6	**74.5**	**73.4**	62.3	61.8	**66.7**	73.8	75.9	**82.7**	**77.7**	76.6	75.2
ours+voting	80.4			78.3			90.8			86.4		

TABLE IV: Average accuracy (%) on *VisDA2017*

N_α	4	5	6	7	8
DIVA [25]	69.8±2.5	68.3±1.6	63.7±3.3	59.0±3.8	57.0±4.5
JiGen [26]	**73.2±2.7**	68.0±5.2	65.9±3.8	**63.7±5.6**	**59.7±2.6**
MATE [24]	68.5±4.0	68.7±3.2	61.4±3.1	56.5±5.6	50.9±4.3
Epi-FCR [27]	72.8±3.4	69.3±5.5	**66.9±5.3**	61.2±4.9	54.9±5.0
MMD-AAE [28]	68.5±5.8	60.4±2.5	59.4±3.3	53.4±2.4	45.0±2.7
CoCoGAN [16]	71.3±1.3	**69.4±2.4**	61.6±2.3	60.9±2.6	56.0±1.7
ZDDA+DC [15]	62.4±2.5	59.7±1.0	61.4±1.7	52.5±1.8	50.9±2.1
Ours	**76.8±1.2**	**74.3±1.4**	**70.4±0.9**	**68.4±1.7**	**64.4±1.2**

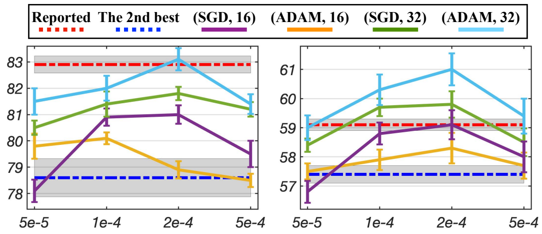
As it is difficult to identify homogeneous datasets for the last two datasets, we transfer the domain shift across categories inside a dataset. The RT involves a subset of the categories and the IRT involves the remaining categories for a specified pair of domains. Let N_α be the number of categories in the RT. Tab. III presents the results with $N_\alpha = 25$ for all twelve possible (source, target) domain pairs on Office-Home. The baselines [24]–[28] use all the samples in their three source domains for training, and their training sets are approximately 1.5 times as large as ours. In terms of average accuracy, our method performs the best (72.77%), followed by Epic-FCR [27] (71.08%) and JiGen [26] (69.28%). If we use a voting method to combine the results from the three source domains, the average accuracy that is realized by our method becomes 83.98%, which significantly exceeds those of all the baselines. Tab. IV presents performance comparisons of all considered methods on VisDA2017 in the adaptation from the synthetic domain to the real domain, with the parameter N_α taking the values of 4, 5, 6, 7, and 8. Our method realizes the highest accuracy and outperforms the baselines by $\geq 3.5\%$ in all five cases. Baseline MATE [24] cannot perform well on these two datasets (in comparison with the first four) since these two datasets are more challenging. In the task of ZSDA from multiple source domains, Epic-FCR [27] and JiGen [26] are the best two choices for the last two challenging datasets, and MATE [24] is the best choice for the first four easier datasets.

On Office-Home, we conduct an ablation study in which we evaluate the contributions of our three key components by replacing them with their alternatives. First, a task classifier can replace GAN^{shift} to realize domain shift transfer (*Xfer*). We can use a task classifier to identify the task label of the representation difference between the *paired samples*, *i.e.,* to discriminate δ_x^r from δ_x^{ir}. Second, the operation between the representations of the *paired samples* can be changed from element-wise subtraction (*i.e.,* $\ominus$) to concatenation for domain

TABLE V: Ablation study on *Office-Home*: "✓" indicates that the component of our method is adopted and " – " indicates that the alternative is adopted.

Xfer	*Shift*	*Init*	*Pr* → *Ar*	*Rw*→ *Cl*	*Rw*→ *Pr*	*Ar* → *Rw*	*Avg.*
–	–	–	48.9±3.7	40.3±2.8	46.0±2.3	45.6±1.2	45.20
–	–	✓	55.6±2.9	50.7±3.5	52.5±3.5	56.3±1.1	53.78
–	✓	–	52.1±3.8	53.0±2.1	59.3±1.3	56.5±2.5	55.23
✓	–	–	54.4±2.9	49.5±1.8	63.3±3.4	57.2±3.0	56.10
–	✓	✓	67.6±1.2	63.4±2.9	73.9±2.7	65.1±2.9	67.50
✓	–	✓	71.4±2.4	58.0±2.6	69.1±3.7	70.5±2.7	67.25
✓	✓	–	70.0±1.2	60.9±2.5	76.4±2.3	71.7±2.1	69.75
✓	✓	✓	74.5±2.5	66.7±1.3	82.7±2.2	77.7±3.8	75.40

shift modeling. Third, we can initialize (*Init*) the non-shared layers of GAN$_2^r$ by a random method instead of copying them from GAN$_2^{ir}$. Tab. V lists the results, where the notations "✓" and " – " indicate the adoption of our components and their alternatives, respectively. These three components can improve the performance complementarily, which justifies their integration into our method. In the extreme case in which we replace all of these three components with their alternatives, the average accuracy drops significantly by 30.2%.

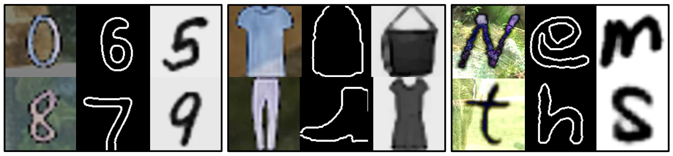


(a) (IRT, RT)=(D_F, D_M) *(b) (IRT, RT)=(D_M, D_F)*

Fig. 7: Robustness study: accuracy *vs.*learning rate under various (optimization method and batch size) settings. We divide the *std* by 4 for visualization.

Considering two (IRT, RT) pairs as examples, namely, (D_M, D_F) and (D_F, D_M), we evaluate the robustness of our method against the variations of the hyperparameters and the optimization methods in the adaptation from G*–dom* to C*–dom*. Here, we train the model with the same optimization method (*ADAM* or *SGD*), learning rate ($5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}$, or 5×10^{-4}), and batch size (16 or 32) at all stages and plot the accuracies in Fig. 7. This figure also shows the reported accuracies that are realized under our default settings (dashed red) and those of the second best model in Tab. II (dashed blue). The second best model is MATE [24] in (a) and Epic-FCR [27] in (b). Our method always outperforms the second best model with a learning rate of either 10^{-4} or 2×10^{-4}. With these learning rates, our method can realize even higher accuracies than the reported accuracies on these two adaptation tasks. In comparison with SGD, ADAM performs slightly better with our model when both the learning rate and the batch size are fixed.

With *CoGAN-R*, we not only learn the models but also synthesize data for the unseen target domain. Regarding G*–dom* as the source domain, Fig. 8 visualizes the generated images in the other three domains. The generated images are in the same style as the real data in the corresponding datasets. Let $(\tilde{x}_s^r, \tilde{x}_t^r)$ be the synthesized image pairs. We quantitatively evaluate *CoGAN-R* in capturing the joint distribution of dual-domain data by calculating the average ratio of the overlapped pixels between $\tilde{x}_t^r$ and $T(\tilde{x}_s^r)$, where $T(\tilde{x}_s^r)$ is the transformation result of $\tilde{x}_s^r$ that is obtained using the method that is detailed in Sec. V-A1. As presented in Tab. VI, the high overlap ratio validates the correspondence between the generated sample pairs and demonstrates the capability of our *CoGAN-R* in capturing the joint distribution of samples in RT. The overlap ratio of the C*–dom* data is meaningless due to the random patch sampling procedure in its creation method.



(a) MNIST *(b) Fashion MNIST* *(c) EMNIST*

Fig. 8: Synthesized images in three domains.

TABLE VI: Average overlap ratios between the generated target-domain images and the images that are obtained via the procedures that are detailed in Sec. V-A1.

	D_M	D_F	D_N	D_E
E–domain	0.943	0.857	0.921	0.906
N–domain	0.961	0.875	0.932	0.894

We also compare GAN_2^r with the independent network GAN_2^{r+} in generating the digit images of C*–dom*, and we use the Fréchet inception distance (FID) [49] as a metric to assess the similarity between the real samples and the generated samples. Both GAN_2^r and GAN_2^{r+} are initialized via the same approach. While GAN_2^r is optimized based on Eq. (6) for domain shift preservation, GAN_2^{r+} is optimized with real target-domain samples based on Eq. (5). After convergence, the FID of our method is only 4.6 lower than that of GAN_2^{r+} (50.7 vs. 55.3). Hence, GAN_2^r performs similarly to GAN_2^{r+} in generating high-quality data samples.

B. Semantic Segmentation

Semantic segmentation is an important task that enables autonomous systems to interact properly with their surroundings [50]. Most semantic segmentation methods receive RGB images as input. Here, we explore the possibility of learning a model for depth images when only RGB images are available. Such a method can enable depth-based systems to provide continuous services in a specified environment to collaborate with or replace RGB-based systems. We realize this by transferring the domain shift between the RGB data and the depth data from a *synthetic-world* domain to the *real-world* domain.

1) Datasets: SceneNetRGBD [57] is a large synthesized dataset for indoor scene analysis. It has $5M$ frames from more than $16K$ room configurations in the training set and $300K$ frames from $1K$ configurations. This dataset provides a triplet of (RGB, depth, semantic label map) for each frame.

NYUv2 [58] and SUN RGB-D [59] are two popular real-world datasets for indoor scene analysis, where the RGB

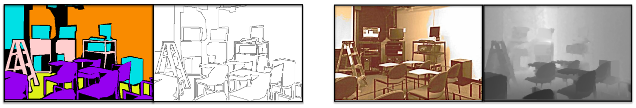
TABLE VII: Results on two datasets. Liu [51]‡ uses unlabeled and Handa Superv. [52]† uses labeled real depth data.

		bed	books	ceil.	chair	floor	furn	objs.	paint.	sofa	table	TV	wall	win.	c-avg.	p-avg.
NYUv2	Handa Superv. [52]†	70.8	$\boxed{5.5}$	76.2	$\boxed{59.6}$	95.9	62.3	50.0	$\boxed{18.0}$	61.3	42.2	$\boxed{22.2}$	86.1	32.1	52.5	67.2
	Handa ZS [52]	**67.9**	4.7	41.2	**67.7**	**87.9**	**38.4**	25.6	6.3	16.3	**43.8**	0.0	**88.6**	1.0	37.6	**55.3**
	ERFNet [53]	51.4	3.5	22.6	31.3	63.2	9.7	21.8	4.6	15.7	16.4	5.7	37.0	5.7	22.0	28.7
	AT/DT [54]	43.7	2.4	31.3	42.1	79.0	27.9	38.2	7.1	32.1	31.8	8.3	66.0	**20.5**	33.1	46.7
	M&MNet [55]	51.4	5.7	25.5	40.6	83.8	17.6	45.1	4.0	13.5	17.5	**19.4**	45.0	10.6	29.2	38.9
	pixel2pixel [56]	60.5	**7.0**	**43.5**	38.1	87.1	37.9	**55.5**	15.4	43.0	25.3	6.2	68.3	26.1	**39.5**	54.0
	Ours*w/o*EM	47.8	4.8	28.5	33.6	60.3	30.0	47.6	7.8	14.4	25.9	12.5	39.9	13.1	28.1	37.3
	Ours$_{clf}$	42.6	2.7	34.9	27.6	76.5	15.5	40.5	**18.4**	**38.5**	33.0	11.1	52.8	20.4	31.9	40.9
	Ours	**70.3**	**10.5**	**53.3**	**64.6**	**89.5**	**46.2**	**49.9**	**24.3**	**54.9**	**47.9**	**22.8**	**86.3**	**27.4**	**49.8**	**63.5**
SUN RGBD	Liu [51]‡	54.1	22.0	47.5	50.4	81.1	36.6	24.8	30.7	46.2	49.2	17.8	70.2	39.0	43.8	60.3
	Handa Superv. [52]†	75.6	$\boxed{13.5}$	69.2	73.6	93.8	52.0	37.1	16.8	57.2	62.7	$\boxed{9.5}$	88.8	36.5	53.1	75.5
	Handa ZS [52]	**46.1**	5.2	43.6	**54.8**	63.1	**37.4**	23.2	10.7	12.2	29.8	0.0	**83.6**	1.0	31.6	**54.8**
	ERFNet [53]	18.5	4.0	14.1	22.1	31.5	9.2	5.9	10.5	5.3	14.9	3.3	27.8	5.7	13.3	21.9
	AT/DT [54]	19.7	**13.1**	25.7	28.3	**78.9**	29.8	15.9	12.9	8.5	22.9	6.0	68.5	**16.8**	26.7	49.3
	M&MNet [55]	27.8	2.1	24.3	39.4	44.5	35.9	**26.7**	**22.6**	15.7	16.8	6.2	53.8	8.4	24.9	38.2
	pixel2pixel [56]	44.6	11.5	39.2	51.7	64.6	34.0	23.8	15.1	**45.3**	20.9	15.4	69.6	9.9	**34.3**	50.8
	Ours*w/o*EM	23.2	2.0	17.8	29.7	48.6	21.8	9.0	12.9	19.1	13.4	6.5	35.2	11.7	19.3	30.6
	Ours$_{clf}$	35.7	9.2	**45.5**	35.0	40.9	28.5	14.9	11.6	28.0	**31.1**	**18.9**	77.8	15.9	30.2	45.6
	Ours	**66.4**	**13.9**	**57.1**	**55.4**	**75.4**	**50.3**	**25.7**	**27.3**	**36.7**	**33.7**	**24.8**	**84.2**	**20.0**	**43.9**	**61.8**

images, the depth images, and the semantic label maps are available. NYUv2 consists of 795 training frames and 654 testing frames. SUN RGB-D consists of $5,285$ training frames and $5,050$ testing frames.

2) Implementation details: The IRT and the RT denote the scene analyses in the synthetic world and the real world, respectively. The source domain consists of the RGB images, and the target domain consists of the depth images. We train *CoGAN-IR* to capture the joint distribution of the available synthetic (RGB, depth) pairs, and we train *CoGAN-R* to predict the joint distribution of the real (RGB, depth) pairs by transferring the domain shift.

In contrast to previously established CoGAN architectures and inspired by [60], we alleviate the training difficulty by feeding the CoGANs two inputs, namely, the semantic label map and its instance-level boundary map, as illustrated in Fig. 9 (a). While the former provides the global structure, the latter differentiates instances of the same label, which can have large depth gaps. Our training procedure uses (label map, depth) pairs in the synthetic domain and (label map, RGB) pairs in both domains. However, we do not use the correspondences between the depth and the RGB in IRT.



(a) Input: label and boundary map (b) Output: RGB and depth image

Fig. 9: Inputs and outputs of the generators in CoGANs

We build the generators in both CoGANs based on the network with skip connections in [61]. The two generators in a CoGAN share the bottleneck layer and the layers before it. To enable the application of our method to this challenging task, we train the *CoGAN-IR* with three stages by feeding the network semantic labels of (i) the patches that consist of a single instance, (ii) the patches that consist of two neighboring instances, and (iii) the whole image. In this way, the network gradually learns how to capture the joint distribution of the (RGB, depth) images from an individual object to complicated scenes and gains the ability to deal with a large depth divergence. GAN^{shift} and GAN$_1^r$ are also trained via this strategy. The hyperparameters are the same as those that are defined in Section V-A for each stage. We feed the trained *CoGAN-R* label maps (for both the whole image and the patch image) to produce a set of depth images. Then, we use these (label map, depth) pairs to fine-tune the ERFNet [53], which is pretrained on the synthetic depth data.

3) Results: We compare our method with eight baselines on semantic segmentation of depth images. In [52], *Handa Superv.* denotes the method that uses both synthetic and labeled real data, and *Handa ZS* denotes the method for zero-shot learning, which does not rely on the target-domain data. *Liu* et al. [51] assume the target domain is available and use paired (depth, RGB) for training. Handa Superv. [52] and the method of Liu [51] are not zero-shot learning methods. In contrast, the benchmarks [53]–[56] and our method do not use real depth data for training. ERFNet [53] directly uses the synthetic data and the RGB images. *AT/DT* [54] derives the unseen depth images and, thus, a segmentation model for depth is learnable. In *M&MNet* [55], while the encoder-decoder for the (RGB, depth) is trained with only the synthetic data, that for (RGB, label map) is trained with both the synthetic data and the real data for fair comparison. Baseline *pixel2pixel* [56] learns an RGB→depth translator based on the synthetic data and uses it to derive depth images from the real RGB images. All the benchmarks [54]–[56] also fine-tune the pre-trained ERFNet [53] with their derived depth images. For an ablation study, we also consider Ours*w/o*EM and Ours$_{clf}$ as the benchmarks. While Ours*w/o*EM removes the boundary map from the inputs of CoGANs, Ours$_{clf}$ replaces GAN$_{shift}$ with a binary task classifier for domain shift alignment.

For both real-world datasets, we segment the depth image into 13 semantic classes and evaluate the performance at 320×240 resolution. Tab. VII lists the accuracy on each class and two metrics for overall assessment, namely, the class average accuracy (c-avg.) and the global pixel average accuracy (p-avg.). ERFNet [53] realizes the lowest accuracy, which provides evidence of the domain shift. In comparison with all the zero-shot learning techniques [54]–[56], our method performs the best and realizes the highest accuracy in 10 classes on both datasets. In terms of pixel average accuracy, our method outperforms Ours*w/o*EM by a margin of 26.2% on NYUv2 and 31.2% on SUN RGBD, thereby demonstrating the effectiveness of the boundary map in guiding the network training. Our method also realizes significant performance gains compared with Ours$_{clf}$. Hence, a GAN network is more suitable than a task classifier for domain shift transfer across tasks. On SUN RGBD, our method even realizes higher overall evaluation metrics than Liu [51] which uses the un-labeled real-data for training. In addition, our method outperforms Handa Superv. [52] in four classes on NYUv2 and two classes on SUN RGBD (Tab. VII).

VI. Conclusions and Future Directions

This paper proposes a new method that provides an effective alternative solution for ZSDA and is based on the hypothesis that different tasks may share the same domain shift for a specified source domain and target domain pair. We learn the domain shift from one task and transfer it to another by bridging two CoGANs with a GAN. Our method defines the domain shift as the distribution of the representation difference between the *paired samples* and models it with a GAN. In comparison with the state-of-the-art methods, our method can not only learn models for the unseen target domain but also generate target-domain data samples. Extensive experiments show that our proposed method is effective in transferring knowledge in both image classification and semantic segmentation in comparison with the state-of-the-art methods.

We identify two directions for further investigation. First, we explore the possibility of transferring the domain shift from an easier IRT to a more challenging RT. To learn a transferable domain shift from an IRT to an RT, our method implicitly assumes that the IRT is similar in difficulty to or more challenging than the RT. However, in practical applications, the RT could be more challenging than the IRT. To address this problem, we will attempt to introduce a self-supervision mechanism to render the IRT more challenging and, hence, simultaneously enhance the transferability of the domain shift. For example, we may rotate the IRT samples and predict both their labels and their rotation angles. Alternatively, we may use unsupervised learning techniques [62] to change the IRT. Second, we will explore freeing our method from dependency on the IRT-data. Our training procedure relies on not only the source-domain samples in the RT but also the dual-domain samples in the IRT. However, the data samples in the IRT could be unavailable in various applications due to privacy concerns in practice (*e.g.,* biometric data). One possible solution is to learn the transferable domain shift based on the pre-trained IRT models instead of the IRT data.

Acknowledgment

The authors wish to acknowledge the financial support from Natural Science Foundation China (NSFC) under the Grant No. 62032015, 61620106008, and 61802266.

References

[1] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, vol. 37, 2015, pp. 1180–1189.

[2] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *ICCV*, 2015, pp. 2452–2460.

[3] Y. Zhang, P. David, H. Foroosh, and B. Gong, "A curriculum domain adaptation approach to the semantic segmentation of urban scenes," *TPAMI*, vol. 42, no. 8, pp. 1823–1841, 2020.

[4] Q. Wang, J. Gao, and X. Li, "Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4376–4386, 2019.

[5] S. Wang, L. Zhang, W. Zuo, and B. Zhang, "Class-specific reconstruction transfer learning for visual recognition across domains," *IEEE Transactions on Image Processing*, vol. 29, pp. 2424–2438, 2020.

[6] M. Tan, J. Yu, H. Zhang, Y. Rui, and D. Tao, "Image recognition by predicted user click feature with multidomain multitask transfer deep network," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6047–6062, 2019.

[7] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar svms," *TPAMI*, vol. 40, no. 5, pp. 1114–1127, 2018.

[8] Y. Liu, Z. Lu, J. Li, T. Yang, and C. Yao, "Deep image-to-video adaptation and fusion networks for action recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 3168–3182, 2020.

[9] S. Bak, P. Carr, and J. Lalonde, "Domain adaptation through synthesis for unsupervised person re-identification," in *ECCV*, vol. 11217, 2018, pp. 193–209.

[10] G. Chen, J. Lu, M. Yang, J. Zhou, and G. Chen, "Learning recurrent 3d attention for video-based person re-identification," *IEEE Transactions on Image Processing*, pp. 1–1, 2020.

[11] S. Motiian, M. Piccirilli, D. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *ICCV*, 2017, pp. 5716–5726.

[12] T. Yao, Y. Pan, C. W. Ngo, H. Li, and M. Tao, "Semi-supervised domain adaptation with subspace learning for visual recognition," in *CVPR*, 2015, pp. 2142–2150.

[13] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *CVPR*, 2019, pp. 5031–5040.

[14] D. Das and C. S. G. Lee, "Graph matching and pseudo-label guided deep unsupervised domain adaptation," in *ICANN*, vol. 11141, 2018, pp. 342–352.

[15] K. C. Peng, Z. Wu, and J. Ernst, "Zero-shot deep domain adaptation," in *ECCV*, 2018, pp. 793–810.

[16] J. Wang and J. Jiang, "Conditional coupled generative adversarial networks for zero-shot domain adaptation," in *ICCV*, 2019, pp. 3374–3383.

[17] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI*, 2018, pp. 3490–3497.

[18] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *ECCV*, vol. 8691, 2014, pp. 628–643.

[19] J. Wang and J. Jiang, "Adversarial learning for zero-shot domain adaptation," in *ECCV*, vol. 12366, 2020, pp. 329–344.

[20] H. Xia and Z. Ding, "Hgnet: Hybrid generative network for zero-shot domain adaptation," in *ECCV*, vol. 12372, 2020, pp. 55–70.

[21] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *TPAMI*, vol. 39, no. 7, 2017.

[22] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *CVPR*, 2020, pp. 12 553–12 562.

[23] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *ICML*, vol. 28, 2013, pp. 10–18.

[24] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *CVPR*, 2015, pp. 2551–2559.

[25] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling, "DIVA: domain invariant variational autoencoders," in *International Conference on Medical Imaging with Deep Learning, MIDL*, vol. 121, 2020, pp. 322–348.

[26] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *CVPR*, 2019, pp. 2229–2238.

[27] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *ICCV*, 2019, pp. 1446–1455.

[28] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *CVPR*, 2018, pp. 5400–5409.

[29] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *ECCV*, vol. 7572, 2012, pp. 158–171.

[30] D. Li, Y. Yang, Y. Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017, pp. 5543–5551.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings IEEE*, 1998.

[32] H. X., K. R., and R. V., "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR2017*.

[33] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *NIPS*, 2016, pp. 469–477.

[34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[35] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *ECCV*, 2018, pp. 647–663.

[36] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *IEEE TIP*, vol. 27, no. 1, pp. 304–313, 2018.

[37] Y. Yang and T. Hospedales, "Zero-shot domain adaptation via kernel regression on the grassmannian," in *DIFF-CV*, 2015.

[38] A. Kumagai and T. Iwata, "Zero-shot domain adaptation without domain semantic descriptors," *ArXiv*, 2018.

[39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[40] P. Grother and K. Hanaoka, "Nist special database 19 handprinted forms and characters database," in *National Institute of Standards and Technology*, 2016.

[41] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *CoRR2017*.

[42] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[43] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017, pp. 5385–5394.

[44] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko, "Visda: A synthetic-to-real benchmark for visual domain adaptation," in *CVPR Workshops*, 2018, pp. 2021–2026.

[45] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016.

[46] O. Sener, H. Song, A. Saxena, and S. Savarese, "Learning transferrable representations for unsupervised domain adaptation," in *NIPS*, 2016, pp. 2110–2118.

[47] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, "Associative domain adaptation," in *ICCV*, 2017, pp. 2784–2792.

[48] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *ICML*, 2017, pp. 2988–2997.

[49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017, pp. 6629–6640.

[50] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang, "Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks," in *ECCV*, vol. 9909, 2016, pp. 664–679.

[51] K. Liu, Y. Shen, J. Klopp, and L. Chen, "What synthesis is missing: Depth adaptation integrated with weak supervision for indoor scene parsing," in *ICCV*, 2019, pp. 7344–7353.

[52] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding realworld indoor scenes with synthetic data," *CVPR*, pp. 4077–4085, 2016.

[53] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE TITS*, vol. 19, no. 1, pp. 263–272, 2018.

[54] P. Z. Ramirez, A. Tonioni, S. Salti, and L. D. Stefano, "Learning across tasks and domains," in *ICCV*, 2019, pp. 8109–8118.

[55] Y. Wang, J. van de Weijer, and L. Herranz, "Mix and match networks: Encoder-decoder alignment for zero-pair image translation," in *CVPR*, 2018, pp. 5467–5476.

[56] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 5967–5976.

[57] J. McCormac, A. Handa, S. Leutenegger, and A. J.Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?" in *ICCV*, 2017, pp. 2697–2706.

[58] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, vol. 7576, 2012, pp. 746–760.

[59] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *CVPR*, 2015, pp. 567–576.

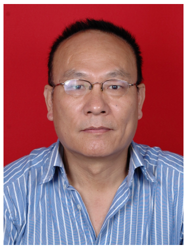
[60] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018, pp. 8798–8807.

[61] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 5967–5976.

[62] J. Wang and J. Jiang, "Unsupervised deep clustering via adaptive GMM modeling and optimization," *Neurocomputing*, vol. 433, pp. 199–211, 2021.

Jinghua Wang received his Ph.D. degree from The Hong Kong Polytechnic University in 2013. From 2014 to 2016, he was a research fellow with Nanyang Technological University. He is currently a Research Assistant Professor with Shenzhen University. His current research interests include domain adaptation, zero-shot learning, computer vision and machine learning.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests includes computer graphics, computer vision, and image processing. He has published 50+ refereed research papers, with 12,000+ Google Scholar citations. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, *etc*. He is a senior member of IEEE and on the editor board of IEEE TIP.



Jianmin Jiang received a PhD from the University of Nottingham, UK, in 1994. From 1997 to 2001, he worked as a full professor of Computing at the University of Glamorgan, Wales, UK. In 2002, he joined the University of Bradford, UK, as a Chair Professor of Digital Media and Director of the Digital Media & Systems Research Institute. He worked at the University of Surrey, UK, as a full professor during 2010-2014 and as a distinguished professor (1000-plan) at Tianjin University, China, during 2010-2013. He is currently a Distinguished Professor and director of the Research Institute for Future Media Computing at the College of Computer Science & Software Engineering, Shenzhen University, China. He was a chartered engineer, fellow of the IEE, fellow of RSA, member of EPSRC College in the UK, and EU FP-6/7 evaluator. His research interests include, image/video processing in compressed domains, digital video coding, medical imaging, computer graphics, machine learning and AI applications in digital media processing, retrieval and analysis. He has published approximately 400 refereed research papers.