Partially-Shared Variational Auto-encoders for Unsupervised Domain Adaptation with Target Shift

Ryuhei Takahashi[1], Atsushi Hashimoto[2], Motoharu Sonogashira[1], and Masaaki Iiyama[1]

[1] Kyoto University, Japan {sonogashira,iiyama}@mm.media.kyoto-u.ac.jp
[2] OMRON SINIC X Corp., Tokyo, Japan atsushi.hashimoto@sinicx.com

Abstract. Target shift, the different label distributions of source and target domains, is an important problem for practical use of unsupervised domain adaptation (UDA); as we do not know labels in target domain datasets, we cannot ensure an identical label distribution between the two domains. Despite this inaccessibility, modern UDA methods commonly try to match the shape of the feature distributions over the domains while projecting the features to labels by a common classifier. This implicitly assumes the identical label distribution. To overcome this problem, we propose a method that generates a pseudo pair by domain conversion where the label is preserved identically even trained with target-shifted datasets. A pair-wise metric learning enables to align feature over the domains without matching the shape of distributions. We conducted two experiments: one is a regression of pose-estimation, where label distribution is continuous and the target shift problem can seriously degrade the quality of UDA. The other is digit classification task where we can systematically control the distribution difference. The code and dataset are available at https://github.com/iiyama-lab/PS-VAEs.

1 Introduction

Unsupervised domain adaptation (UDA) is one of the most studied topics in recent years. One attractive application of UDA is adaptation from computer graphic (CG) data to sensor-observed data. By constructing a CG-rendering system, we can easily obtain a large amount of supervised data with diversity for training. Because a model straightforwardly trained on CG-rendered dataset hardly works with real observation, training a model with both CG-rendered dataset (source domain) and unsupervised real observation dataset (target domain) by UDA is a necessary but promised approach.

As in ADDA [38], the typical approach for UDA is to match feature distributions between the source and target domains [8, 19, 22]. This approach works impressively with identically-balanced datasets, such as those for digits (MNIST, USPS, and SVHN) and traffic-scene semantic-segmentation (GTA5 [30] to Cityscapes [5]). When the prior label distributions of the source and target domains are mismatched, however, such approaches hardly work without a

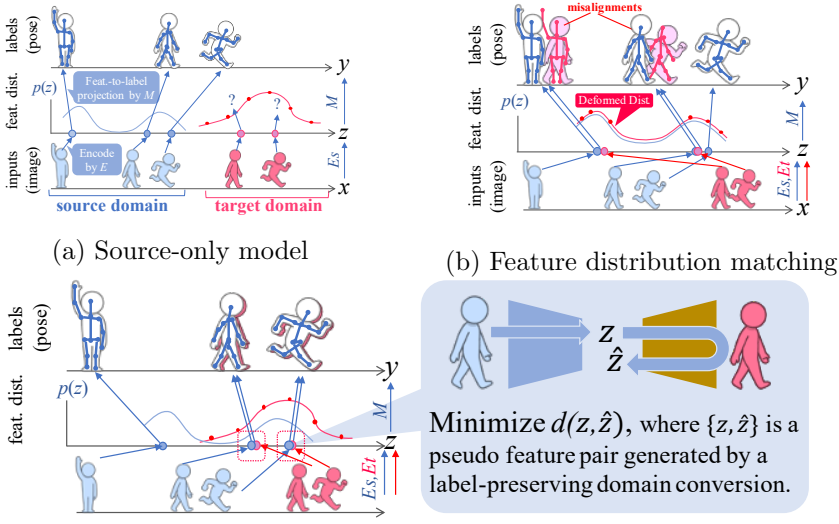


Fig. 1. (best viewed in color) Overview of the proposed approach on the problem of (2D) human pose estimation. Note that the feature-to-label projection M is trained only with source domain dataset, where E_s and E_t are domain specific image-to-feature encoders. (a) The naive approach fails in the target domain due to the differences in the feature distributions between the two domains: location difference that illustrates the affection by domain shift and shape difference caused by *target shift* (non-identical label distributions of the two domains). (b) While feature distribution matching attempts to adjust the shape of two feature distributions, it suffers from misalignment in label estimation due to the deformed target domain feature distribution. (c) The proposed method avoid this deformation problem by sample-wise feature distance minimization, where pseudo sample pairs with an identical label are generated via a CycleGAN-based architecture.

countermeasure for the mismatch (see Figure 1). Cluster finding [34, 33, 6] is another approach for UDA by a class-boundary adjustment; they are not, however, applicable to regression problems due to the absence of class-boundaries in the distribution.

In this paper, we propose a novel UDA method applicable especially to the regression problem with mismatched label distributions. The problem of mismatched label distributions is also known as *target shift* [10, 42] or *prior probability shift* [28]. The typical example of this problem is UDA from a balanced source domain dataset to an imbalanced target domain dataset. Some recent studies have tried to overcome this problem by estimating category-wise importance labels [1–3, 39, 41] or sample-wise importance labels [15]. The former approach is only applicable to classification tasks. The latter is applicable to regression but under-samples the source domain data. In addition, it requires a reliable similarity metric over the domain shift (a domain-shift-free metric) to se-

lect important samples; this suffers from the chicken-and-egg situation. Namely, if we have a measure that is hardly affected by domain shift, we can safely apply pair-wise metric learning (e.g., with a Siamese loss), but such metric is not given in general.

In contrast, our method resolves this problem by oversampling with label-preserving data augmentation. This is applicable even to regression with UDA and does not require any preliminary knowledge of the domain-shift-free metrics. Figure 1 shows the overview. Traditional methods [11, 19, 32, 38] matches feature distributions of the two domains (Fig. 1 (b)). Since feature distributions are forced to be identical and the feature-to-label projection function is shared by the two domains, the estimated labels in the target domain must distribute identically with that of the source domain. Under the target shift condition, this clearly competes with the assumption of non-identical label distributions.

Our method addresses the problem of target shift by tolerating feature distribution mismatches and instead requiring the sample-wise matches of the labels (Fig. 1 (c)). To this end, our method called partially-shared variational auto-encoders (PS-VAEs) organizes a CycleGAN architecture [44] with two VAE branches that share weights as much as possible to realize the label-preserving conversions.

The contribution of this paper is three-fold.

- We propose a novel UDA method that overcomes the target shift problem by oversampling with label-preserving data augmentation, which is applicable to regression. This is the first algorithm that solves regression with UDA under a target shift condition without relying on any prior knowledge of domain-shift-free metrics.
- We tackled the problem of human-pose estimation by UDA with target shift for the first time and outperformed the baselines with a large margin.
- The proposed method showed the versatility under various levels of target shift intensities and different combinations of datasets in the task of digit classification with UDA.

2 Related Work

UDA by a feature space discriminator

The most popular approach in modern UDA methods is to match the feature distributions of the source and target domains so that a classifier trained with the source domain dataset is applicable to target domain samples. There are various options to match the distributions, such as minimizing MMD [22, 39], using a gradient-reversal layer with domain discriminators [8], and using alternative adversarial training with domain discriminators [2, 16, 19, 38]. Adversarial training removes domain bias from the feature representation. To preserve information in domain invariant features as much as possible, UFDN [19] involves a VAE module [7] with the discriminator. Another approach is feature whitening [31], which whitens features from each domain at domain-specific alignment layers. This approach does not use adversarial training, but it tries to analytically fit a feature

Table 1. Representative UDA methods and their supporting situations. The symbol "(✓)" indicates that the method theoretically supports the situation but this was not experimentally confirmed in the original paper.

	Balance		Imbalance	
	classification	regression	classification	regression
ADDA[38], UFDN[19], CyCADA[11]	✓	(✓)		
MCD[34]	✓		(✓)	
PADA[39], UAN[40], CDAN-E[23]	✓		✓	
SimGAN[35]		✓		(✓)
Ours	✓	(✓)	✓	✓

distribution from each domain to a common spherical distribution. As shown in Table 1, all these methods are theoretically applicable to both classification and regression, but it is limited to the situations without target shift.

Cluster finding approaches

MCD was proposed by Saito *et al.* [33, 34], which does not use distribution matching. Instead, the classifier discrepancy is measured based on the difference of decision boundaries between multiple classifiers. DIRT-T [36] and CLAN [24] are additional approaches focusing on the boundary adjustment. These approaches are potentially be robust against target shift, because they focus only on the boundaries and do not try to match the distributions. CAT [6] is a plug-and-play method that aligns clusters found by other backbone methods. Since these approaches assume an existence of boundaries between clusters, they are not applicable to regression, which have continuous sample distributions (see the second row in Table 1).

UDA with target shift

Traditional UDA benchmarks barely discuss the problem of target shift. Most classification datasets, such as MNIST, USPS, and SVHN are balanced. Even the class-imbalance problem is known with semantic segmentation, target shift does not come to the surface as long as source and target domains are similarly imbalanced (i.e., their label distributions can be considered as identical). GTA5→Cityscapes is in the case. CDAN-E [23] is one of the few methods that has potential to deal with target shift although the original paper does not clearly discuss the target shift problem. Partial domain adaptation (PDA) is a variant of UDA with several papers on it [1–3, 39, 41] (see the third row in Table 1). This problem assumes a situation in which some categories in the source domain do not appear in the target domain. This problem is a special case of target shift in two senses: it assumes the absence of a category and it assumes only classification tasks. The principle approach for this problem is to estimate the importance weight for each category, and ignore those judged as unimportant (under-sampling). UAN [40] is another recent method that solves PDA and the open-set problem simultaneously. It estimates sample-wise importance weight based on the entropy at the classification output for each target sample. PADACO [15] is designed for a regression problem of head pose estimation under a target shift situation. To obtain sample-wise importance weight with a regres-

sion problem, it uses head-pose similarity between source and target samples, where target domain head-pose is estimated by a pretrained backbone, which is source-only model in the paper. Then, the similarity values are converted into fixed sampling weights (under-sampling). Finally, it performs UDA training with a weighted sampling strategy. To obtain better results with this method, it is important to obtain good sampling weights with the backbone, just as CAT [6].

UDA by domain conversion

Label-preserving domain conversion is another important approach and includes the proposed method (see fourth and fifth rows in Table 1). Shrivastava *et al.* proposed SimGAN [35], which converts CG images to nearly real images by adversarial training. This method tries to preserve labels by minimizing the self-regularization loss, the pixel-value difference between images before and after conversion. This method can be seen as an approach based on over-sampling with data augmentation in the sense that it generates source-domain-like samples using GAN under the self-regularization constraint. Note that SimGAN is the first deep-learning-based UDA method for regression that is theoretically applicable to the task with target shift. On the other hand, this method still assumes a domain-shift-free metric of the self-regularization loss, which is not always domain-shift-free.

CyCADA [11] combines CycleGAN, ADDA and SimGAN for a better performance. It first generates fake target domain images via CycleGAN. The label-consistency of generated samples are preserved by SimGAN's self-regularization loss; however it has a discriminator that matches the feature distributions. Hence, this methods principally has the same weakness against target shift. SBADA-GAN [32] is yet another CycleGAN-based method with discriminator for feature distribution matching.

In addition to the above methods, there is a recent attempt to solve human-pose estimation by domain adaptation [43].This method tried to regularize domain difference by sharing a discrete space of body-parts segmentation as an intermediate representation, but the reported score shows that the method dose not work effectively under the UDA setting.

3 Method

3.1 Problem statement

Let $\{x_s, y_s\} \in X_s \times Y_s$ be samples and their labels in the source domain dataset (Y_s is the label space), and let $x_t \in X_t$ be samples in the target domain dataset. The target labels Y_t and their distribution $\Pr(Y_t)$ are unknown (i.e., possibly $\Pr(Y_t) \neq \Pr(Y_s)$) in the problem of UDA with target shift. The goal of this problem is to obtain a high-accuracy model for predicting the labels of samples obtained in the target domain.

3.2 Overview of the proposed method

The main strategy of the proposed method is to replace the feature distribution matching process with pair-wise feature alignment. To achieve this, starting from

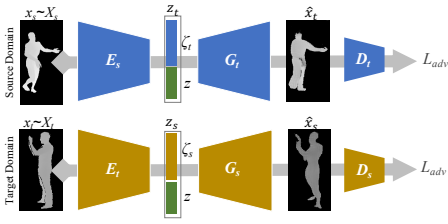


Fig. 2. The basic CycleGAN architecture. We abbreviated the identity loss (L_{id}) and cycle consistency loss (L_{cycle}) for the simplicity.

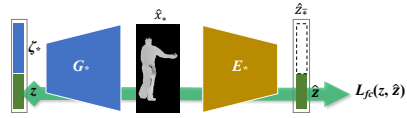


Fig. 3. Feature consistency loss (for both directions $s \to t$ and $t \to s$).

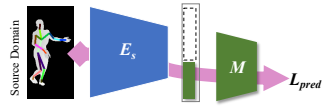


Fig. 4. The prediction loss, which is calculated only with source domain samples.

the standard CycleGAN as the base architecture (Fig. 2), we add two new losses to generate pseudo pairs $(x_s, \hat{x}_t)$ and $(x_t, \hat{x}_s)$, each of which are expected to have an identical label: L_{fc} for feature alignment (Fig. 3) and L_{pred} for label prediction (Fig. 4). The both losses are calculated only on the domain-invariant component z of the disentangled feature representation z_s (or z_t). After the training, prediction in target domain is done by the path, encoder→predictor $(M \circ E_t)$. 3.3 describes this modification in detail.

To preserve the label-related content at domain conversion, we further modify the network by sharing weights and introducing VAE's mechanism (Fig. 5). 3.4 describes this modification in detail.

3.3 Disentangled CycleGAN with feature consistency loss

The model in Figure 2 has pairs of encoders E_*, generators G_*, and discriminators D_*, where $* \in \{s, t\}$. $\hat{x}_t$ is generated by $G_t(E_s(x_s))$, and $\hat{x}_s$ by $G_s(E_t(x_t))$. The original CycleGAN [44] is trained by minimizing the cycle consistency loss L_{cyc}, the identity loss L_{id}, and the adversarial loss L_{adv} defined in LSGAN [26]:

$$\min_{E_s,E_t,G_s,G_t} L_{cyc}(X_s, X_t) = \sum_{*\in\{s,t\}} \mathbb{E}_{x\in X_*}[d(x, \hat{\hat{x}})], \tag{1}$$

where d is a distance function and $\hat{\hat{x}} = G_*(E_{\bar{*}}(\hat{x}_{\bar{*}}))$. Here,$\bar{*}$ is the opposite domain of $*$.

$$\min_{E_s,E_t,G_s,G_t} L_{id}(X_s, X_t) = \sum_{*\in\{s,t\}} \mathbb{E}_{x\in X_*}[d(x, G_*(E_*(x)))] \tag{2}$$

$$\min_{E_s,E_t,G_s,G_t} \max_{D_s,D_t} L_{adv}(X_s, X_t) = \mathbb{E}_{\{x_s,x_t\}\in X_s\times X_t}[\|D_s(x_s) - 1\|_2 + \|D_s(G_s(E_t(x_t))) + 1\|_2 + \|D_t(x_t) - 1\|_2 + \|D_t(G_t(E_s(x_s))) + 1\|_2] \tag{3}$$

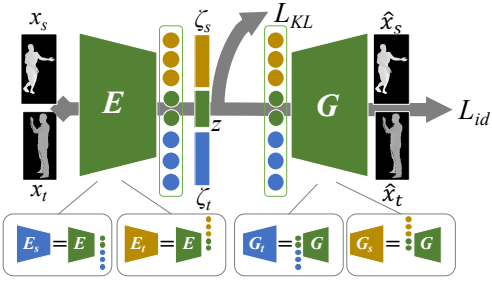


Fig. 5. (best viewed in color) Architecture of partially-shared variational auto-encoders. We note that x_s/x_t is input to E_t/E_s to calculate L_{id}. For a label-preserving domain conversion, the encoders and decoders share parameters other than the connection to ζ_*.

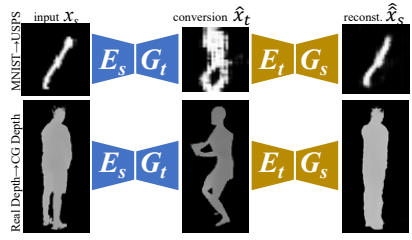


Fig. 6. Misalignment caused by CycleGAN's two image-space discriminators. This is typically seen with a model that does not share encoder weights.

Note that we used spectral normalization [27] in D_s and D_t for a stable adversarial training.

To successfully achieve pair-wise feature alignment, the model divides the output of E_* into $z_{\bar{*}} = \{z, \zeta_{\bar{*}}\}$. Then, it performs feature alignment by using the domain-invariant feature consistency loss L_{fc} (Fig. 3), defined as

$$\min_{E_s,E_t,G_s,G_t} L_{fc}(Z_s, Z_t) = \sum_{*\in\{s,t\}} \mathbb{E}_{z_*\in Z_*}[d(select(z_*), select(E_{\bar{*}}(G_{\bar{*}}(z_*))))], \quad (4)$$

where $Z_* = E_*(X_*)$ and *select* is a function to select z in z_*. Note that gradients are not further back-propagated to $E_{\bar{*}}$ over z_* because updating both z_* and $\hat{z}_*$ in one step leads to bad convergence.

In addition, z obtained from x_s is fed into M to train the classifier/regressor $M : z \to \hat{y}$ by minimizing the prediction loss $L_{pred}(X_s, Y_s)$ (Fig. 4). The concrete implementation of L_{pred} is task-dependent.

We avoid applying L_{fc} to the whole feature components z_t, as it can hardly reach good local minima because of the competition between the pair-wise feature alignment (by L_{fc}) and CycleGAN (by L_{cyc} and L_{id}). Specifically, training G_t to generate $\hat{x}_t$ must yield a dependency of $\Pr(z_t|x_t)$. This means that $\hat{z}_t$ is trained to have in-domain variation information for x_t. The situation is the same with x_s and z_s. Hence, $\hat{z}_t$ and $\hat{z}_s$ have dependencies on different factors, x_t and x_s, respectively, and it is difficult to match the whole features, $\hat{z}_t$ and $\hat{z}_s$. The disentanglement into z and ζ_* resolves this situation. Note that this architecture is similar to DRIT [18] and MUNIT [12].

3.4 Partially shared VAEs

Next, we expect E_s and E_t to output a domain-invariant feature z. Even with this implementation, however, CycleGAN can misalign an image's label-related

content in domain conversions under a severe target shift, because it has discriminators that match not feature- but image-distributions. Figure 6 shows actual examples of misalignment caused by the image space discriminators. This happens because the decoders G_s and G_t can convert identical zs into different digits (or poses) to better minimize L_{adv} with mismatched label distributions. In such cases, the corresponding encoders also extract identical zs from images with totally different appearance.

To prevent such misalignment and get more stable results, we make the decoders share weights to generate similar content from z, and we make the encoders extract z only from similar content. Figure 5 shows the details of the parameter-sharing architecture, which consists of units called *partially-shared auto-encoders* (PS-AEs). Formally, the partially shared encoders are described as a function $E : x \to \{z, \zeta_s, \zeta_t\}$. In our implementation, only the last layer is divided into three parts, which outputs z, ζ_s, and ζ_t. E can obviously be substituted for E_s and E_t by discarding ζ_t and ζ_s from the output, respectively. Similarly, the generator $G : \{z, \zeta_s, \zeta_t\} \to \hat{x}$ shares weights other than for the first layer. The first layer consists of three parts, which output z, ζ_s, and ζ_t. G can be substituted for G_s and G_t by inputting $\{z, \zeta_s, \mathbf{0}\}$ and $\{z, \mathbf{0}, \zeta_t\}$, respectively. Note that the reparameterization trick and L_{kl} minimization are applied only at L_{id} calculation, but not at L_{cycle} calculation.

This implementation brings another advantage for UDA tasks: it can disentangle the feature space by consisting of two variational auto-encoders (VAEs), $G_s \circ E_t$ and $G_t \circ E_s$ (Figure 5). Putting VAE in a model to obtain a domain-invariant feature is reported as an effective option in recent domain adaptation studies [19, 21]. To make PS-AEs a pair of VAEs, we put VAE's resampling process at calculation of L_{id} and add the KL loss defined as

$$\min_{E_s, E_t} L_{KL}(X_s, X_t) = \sum_{*\in\{s,t\}} \mathbb{E}_{z,\zeta_* \in E_*(X_*)}[KL(p_z||q_z) + KL(p_{\zeta_*}||q_{\zeta_*})], \tag{5}$$

where $KL(p\|q)$ is the KL divergence between two distributions p and q, p_{ζ_*} is the distribution of ζ_* sampled from X_*, and q_z and q_{ζ_*} are standard normal distributions with the same sizes as z and ζ_*, respectively.

Our full model, *partially-shared variational auto-encoders* (PS-VAEs), is trained by optimizing the weighted sum of the all the above loss functions:

$$L_{total} = L_{adv} + \alpha L_{cyc} + \beta L_{id} + \gamma L_{KL} + \delta L_{fc} + \epsilon L_{pred}, \tag{6}$$

where α, β, γ, δ, and ϵ are hyper-parameters that should be tuned for each task. For the distance function d, we use the smooth L1 distance [9], which is defined as

$$d(a, b) = \begin{cases} \|a - b\|_2 & \text{if } |a - b| < 1 \\ |a - b| - 0.5 & \text{otherwise} \end{cases} \tag{7}$$

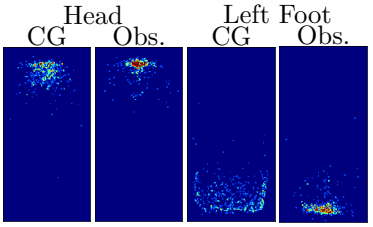


Fig. 7. Difference in joint position distributions. A complete list appears in the supplementary materials.

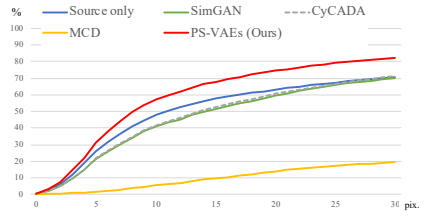


Fig. 8. (best viewed in color) Averaged percentage of joints detected with errors less than N pixels. (Higher is better.)

4 Evaluation

4.1 Evaluation on human-pose dataset

We firstly evaluated the proposed method with a regression task on human-pose estimation.

Dataset For this task, we prepared a synthesized depth image dataset whose poses were sampled with CMU Mocap [4] and rendered with PoserPro2014 [37], as the source domain dataset. Each image had 18 joint positions. In the sampling, we avoided pose duplication by confirming that at least one joint had a position more than 50mm away from its position in any other samples. The total number of source domain samples was 15000. These were rendered with a choice of two human models (male and female), whose heights were sampled from a normal distribution with respective means of 1.707 and 1.579m and standard deviations of 56.0mm and 53.3mm). For the target dataset, we used depth images from the CMU Panoptic Dataset [14], which were observed with a Microsoft Kinect. We automatically eliminated the background in the target domain data by preprocessing.[3] Finally, we used 15,000 images for training and 500 images (with manually annotated joint positions) for the test.

Experimental Settings Figure 7 shows the target shift between the source and target domains via the differences in joint positions at the head and foot. L_{pred} was defined as

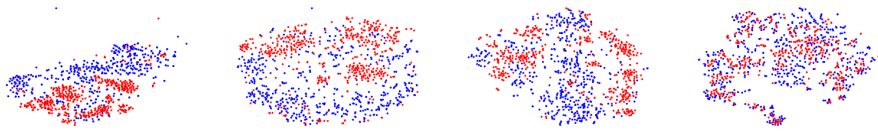
$$\min_{E_s,M} L_{pred}(Y_s, X_s) = \mathbb{E}_{x_s,y_s\in\{X_s,Y_s\}}(d(M(E_s(x_s)), y_s)). \tag{8}$$

Comparative methods We compared the proposed method with the following three baselines.

[3] The details appear in the supplementary material.

Table 2. Accuracy in human-pose estimation by UDA (higher is better). Results were averaged for joints with left and right entries (e.g., the "Shoulder" column lists the average scores for the left and right shoulders). The "Avg." column lists the average scores over all samples, rather than only the joints appearing in this table.

Error less than 10px.	Head	Neck	Chest	Waist	Shldr.	Elbow	Wrists	Hands	Knees	Ankles	Foots	Avg.
Source only	69.6	78.6	31.6	34.2	47.3	44.5	38.4	31.5	38.5	54.1	66.2	47.4
MCD	4.6	7.0	0.2	0.6	1.4	0.2	0.3	0.9	0.4	21.0	16.6	5.3
SimGAN	90.2	68.0	10.8	22.6	38.8	26.3	28.5	33.6	35.9	52.5	52.8	40.4
CyCADA	90.0	69.0	15.4	28.2	39.5	27.3	31.3	32.5	35.4	54.4	53.2	41.0
Ours												
CycleGAN+L_{fc}	82.8	79.0	33.8	17.0	40.0	16.4	15.8	28.4	13.8	51.0	51.5	35.5
D-CycleGAN	**93.0**	**85.8**	21.4	**47.8**	42.5	42.5	35.8	39.2	42.5	66.9	64.1	50.8
D-CycleGAN+VAE	40.6	34.2	17.6	41.2	10.1	10.2	7.5	6.4	20.0	28.0	20.2	18.6
PS-AEs	80.6	72.4	**40.8**	28.0	46.5	28.4	25.2	29.4	25.3	58.9	53.9	42.1
PS-VAEs(full)	89.4	84.6	21.4	43.4	**51.7**	**54.4**	**49.4**	**43.9**	**45.6**	**74.5**	**74.0**	**57.0**



(a) Source Only (b) SimGAN (c) CyCADA (d) Ours

Fig. 9. (best viewed in color) Feature distribution visualized by t-SNE [25]: source domain CG data (blue points) and target domain observed data (red points).

SimGAN [35] is a method based on image-to-image conversion. To prevent misalignment during conversion, it also minimizes changes in the pixel-values before and after conversion by using a self-regularization loss. The code is borrowed from the implementation of CyCADA.

CyCADA [11] is a CycleGAN-based UDA method. The self-regularization loss is used in this method, too. In addition, it matches feature distributions, like ADDA.

MCD [34] is a method that minimizes a discrepancy defined by the boundary differences obtained from multiple classifiers. This method is expected to be more robust against target shift than methods based on distribution matching, because it does not focus on the entire distribution shape. On the other hand, this kind of approach is theoretically applicable only to classification but not to regression.

All the above methods were implemented with a common network structure, which appears in the supplementary materials.

In addition to the above methods, on the purpose of an ablation study, we compared our full model with the following four different variations.

CycleGAN+L_{fc} does not divide z_s and z_t into the two components, but applied L_{fc} to z_s and z_t directly.

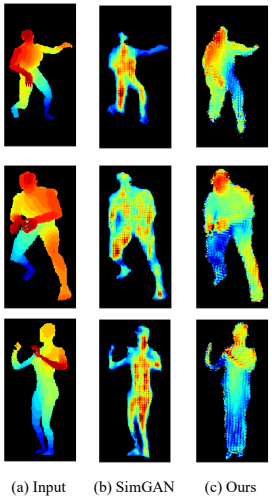


Fig. 10. (best viewed in color) Qualitative comparison on the domain conversion. Detailed structure in body region is lost with SimGAN, but reproduced with our model.

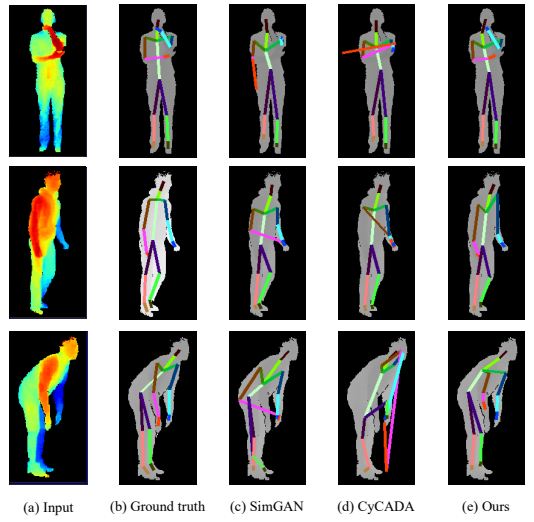


Fig. 11. (best viewed in color) Qualitative results of human-pose estimation. Due to the lack of detailed depth structure as seen in Fig. 10, SimGAN and CyCADA often fail to estimate joints with self-occlusion.

D-CycleGAN stands for disentangled CycleGAN, which divides z_s and z_t into the two components, but weights of encoders and decoders are not shared and not using VAE at the calculation of L_{id}.

D-CycleGAN+VAE is a D-CycleGAN with L_{KL} and the resampling trick of VAE at the calculation of L_{id}.

PS-AEs stands for Partially-shared Auto-Encoders, whose encoders and decoders partially shares weights as described in 3.4, but not using VAE.

PS-VAEs stands for Partially-shared Variational Auto-Encoders and this is the full model of the proposed method.

All hyper-parameters of baselines and the proposed methods are manually tuned with our best effort for this new task. (A comparison under the optimal hyper-parameter settings are given in 4.2 with the other task.)

Results and Comparison Figure 8 shows the rate of samples whose estimated joint position error is less than thresholds (the horizontal axis shows the threshold in pixels). To view the joint-wise tendency, we trimmed the figure at the threshold of ten pixels and listed joint-wise accuracy in Table 2.The full model using the proposed methods achieved the best scores on average and for all the joints other than the head, neck, chest, and waist. These four joints have less target shift than others do (see Figure 7 or the complete list of joint position

distributions in supplementary materials). SimGAN was originally designed for a similar task (gaze estimation and hand-pose estimation) and achieved better scores than MCD. CyCADA is an extension of SimGAN and has additional losses for distribution matching, but it did not boost the accuracy in the tasks of UDA with target shift. MCD was originally designed for classification tasks and did not work for this regression task, as expected.

Discussion Figure 9 shows the feature distributions obtained from four different methods. Because SimGAN does not have any mechanisms to align features in the feature space, the distributions did not merge well. CyCADA better mixes the distributions, but still the components are separated. In contrast, the proposed method merged features quite well despite no discriminators or discrepancy minimization was performed. This indicates that the proposed pair-wise feature alignment by L_{fc} worked well with this UDA task.

A qualitative difference in domain conversion is shown in Figure 10. SimGAN's self-regularization loss worked to keep the silhouette of generated samples, but subtle depth differences in the body regions were not reproduced well. In addition, the silhouettes were forced to be similar to those of the two human models used to render the CG dataset. This insists that the prior assumption of domain-shift-free metric (i.e., self-regularization loss) could rather reduce the accuracy from source only model. In contrast, the proposed method seems to be able to reproduce such subtle depth differences with a more realistic silhouette. This difference contributed to the prediction quality difference shown in Figure 11.

D-CycleGAN had actually performed the second best result and D-CycleGAN+VAE and PS-AEs did not work well. First, as UNIT [20] does,[4] it seems to be difficult to use VAE with CycleGAN without sharing weights between the encoder-decoder models. After combining all these modifications, the full model of the proposed method outperformed any other methods with a large margin.

4.2 Evaluation on digit classification

To show the versatility of the proposed method with classification task and to systematically analyze the performance of the methods against target shift, we conducted an experiment by a simple UDA task with digit datasets (MNIST[17]↔USPS[13], and SVHN[29]→MNIST), with which the optimal hyper-parameters are provided in many methods.

Controlling the intensity of target shift To evaluate the performance under a controlled situation with an easy-to-reproduce and high-contrast class-imbalances in the target domain, we adjusted the rate of samples of class '1' from

[4] Another neural network model that combines CycleGAN and VAE as the proposed model, but for image-to-image translation.

Table 3. Accuracy in the three UDA tasks (Bold&Underline: The best and second best scores. Δ: the degradation from 10% to 50%).

		Src. Only	ADDA	UFDN	CDAN-E	PADA	UAN	SimGAN	CyCADA	MCD	Ours
MNIST→USPS	Ref.	-	89.4	97.1	95.6	-	-	-	95.6	94.2	-
	10%	71.0	89.8	**94.0**	91.0	75.3	78.2	72.4	91.8	91.2	93.9
	20%	-	86.9	90.4	90.7	77.7	77.3	86.5	91.0	90.4	**94.8**
	30%	-	79.3	83.2	91.1	79.3	74.9	84.0	80.3	79.0	**93.4**
	40%	-	81.8	82.3	84.2	77.8	75.0	84.3	86.4	78.5	**94.6**
	50%	-	78.5	83.8	79.8	80.2	76.0	76.3	87.6	80.3	**92.6**
	Δ	-	-11.3	-10.2	-11.2	4.9	-2.2	3.9	-4.2	-10.9	-1.3
USPS→MNIST	Ref.	-	90.1	93.7	98.0	-	-	-	96.5	94.1	-
	10%	55.6	**96.0**	93.6	95.8	47.9	83.2	68.3	75.3	**96.0**	94.8
	20%	-	89.0	81.9	**95.8**	39.2	83.4	50.2	75.3	81.5	94.4
	30%	-	81.5	79.2	**95.4**	36.0	79.4	49.9	75.2	79.1	90.8
	40%	-	78.9	72.0	**90.9**	29.8	78.4	63.8	76.7	78.1	82.6
	50%	-	80.5	69.1	**90.7**	25.2	77.7	49.3	70.7	77.4	82.4
	Δ	-	-15.5	-24.6	-4.1	-22.7	-5.5	-19.0	-4.6	-18.4	-12.4
SVHN→MNIST	Ref.	-	76.0	95.0	89.2	-	-	-	90.4	96.2	-
	10%	46.6	75.5	91.1	78.7	30.5	68.0	61.4	**91.4**	90.3	73.7
	20%	-	65.0	70.9	79.4	39.5	64.8	52.5	75.4	**89.7**	72.9
	30%	-	65.2	58.7	73.2	37.3	63.2	57.7	69.7	**80.2**	73.8
	40%	-	50.8	52.6	56.9	36.8	64.9	51.8	70.7	**72.0**	64.4
	50%	-	54.3	43.6	56.1	36.7	66.8	49.3	68.3	65.3	**68.4**
	Δ	-	-21.2	-47.5	-22.6	6.2	-1.2	-12.1	-23.1	-25.0	-5.3

10% to 50%. Note that the operation is done only to the training samples in the target domain. Those in the source domain and test data are both maintained to be balanced.

When the rate was 10%, the number of samples was exactly the same among the categories. When it was 50%, half the data belonged to category '1,' which was the largest target shift in this experiment. Note that the reference scores reported in the original papers and those at 10% are slightly different due to the controled numbers of training data. A more detailed explanation of this operation appears in the supplementary materials.

Experimental settings and comparative methods In this task, L_{pred} is simply given as the following categorical cross-entropy loss:

$$\min_{E_s,M} L_{pred}(Y_s, X_s) = \mathbb{E}_{x_s,y_s \in X_s \times L}[-y_s \log M(E(x_s))] \tag{9}$$

In addition to the comparative methods shown in 4.1, we prepared the following three additional baselines as the methods that resolve domain shift purely by distribution matching:

ADDA [38] and UFDN [19] are methods based on feature distribution matching.

PADA [2] and UAN [40] also match feature distributions but while estimating a category- and sample-wise weights, respectively.

CDAN-E [23] uses category-wise distribution matching and thus potentially valid under target shift as long as the target domain samples are assigned to the right category. To ensure the right assignment, the method estimates sample-wise weights.

Note that some of recent state-of-the-art methods for the digit UDA task without target shift was not listed in the experiment due to their reproducibility problem.[5] The detailed implementations (network architecture, hyperparameters, and so on) of the proposed method and the above methods appear in the supplementary material.

Results and discussion Table 3 lists the results. The methods based on distribution matching (ADDA and UFDN) were critically affected by target shift. CyCADA was more robust than ADDA and UFDN for the MNIST↔USPS tasks, owing to the self-regularization loss; however it did not work for the SVHN→MNIST task due to the large pixel-value differences. A similar tendency was observed by SimGAN.

MCD stably performed well among all the three tasks; however, it was largely affected by target shifts ($\geq$30%). Similar tendency was observed with CDAN-E ($\geq$40%). From the perspective of the performance drop by target shift, PADA behaved differently from any other methods; it typically works better with a heavier target shift but not so good without target shift. UAN, which was not evaluated on this dataset in the original paper, achieved a poor absolute performance although it was least degraded by the target shift. Overall, our method comparably performed under the various level of target-shifted conditions even in the classification task. This shows the versatility of the method against various UDA tasks.

5 Conclusion

In this paper, we have proposed a novel approach of partially-shared variational auto-encoders for the problem of UDA with target shift. The traditional approach of feature distribution matching implicitly assumes the identical distribution and will fail with target shift. The proposed method resolves this problem by label-preserving domain conversion; pseudo pair with an identical label is generated with domain conversion and used to resolve domain shift by sample-wise metric learning rather than a distribution matching. The model is specially designed to preserve the labels by sharing weights between two domain conversion branches as much as possible. The experimental results showed its versatile performance on pose estimation and digit classification tasks.

[5] The authors of [31] provide no implementation and there are currently no other authorized implementations. Two SBADAGAN[32] implementations were available but it was difficult to customize them for this test and the reported accuracy was not reproducible.

References

1. Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (2018)
2. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: The European Conference on Computer Vision (2018)
3. Cao, Z., You, K., Long, M., Wang, J., Yang, Q.: Learning to transfer examples for partial domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)
4. CMU Graphics Lab.: CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/, (accessed on 11th-Nov-2019)
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)
6. Deng, Z., Luo, Y., Zhu, J.: Cluster alignment with a teacher for unsupervised domain adaptation. In: The IEEE International Conference on Computer Vision (2019)
7. Doersch, C.: Tutorial on variational autoencoders. In: CoRR (2016)
8. Ghifary, M., Kleijn, W.B., Zhang, M.: Domain adaptive neural networks for object recognition. In: Pacific Rim International Conference on Artificial Intelligence. pp. 898–904. Springer (2014)
9. Girshick, R.: Fast R-CNN. In: The IEEE International Conference on Computer Vision (2015)
10. Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., Schölkopf, B.: Domain adaptation with conditional transferable components. In: International Conference on Machine Learning. pp. 2839–2848 (2016)
11. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Proceedings of the 35th International Conference on Machine Learning (2018)
12. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: The European Conference on Computer Vision (2018)
13. Hull, J.J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(5), 550–554 (1994)
14. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: The IEEE International Conference on Computer Vision (2015)
15. Kuhnke, F., Ostermann, J.: Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces. In: The IEEE International Conference on Computer Vision (2019)
16. Laradji, I., Babanezhad, R.: M-ADDA: Unsupervised domain adaptation with deep metric learning. In: Proceedings of the 36th International Conference on Machine Learning Workshop (2018)
17. LeCun, Y., Cortes, C.: MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/ (2010)
18. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: The European Conference on Computer Vision (2018)

19. Liu, A.H., Liu, Y.C., Yeh, Y.Y., Wang, Y.C.F.: A unified feature disentangler for multi-domain image translation and manipulation. In: Advances in Neural Information Processing Systems 31. pp. 2590–2599 (2018)
20. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30, pp. 700–708 (2017)
21. Liu, Y., Wang, Z., Jin, H., Wassell, I.: Multi-task adversarial network for disentangled feature learning. In: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3743–3751 (2018)
22. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. pp. 97–105 (2015)
23. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1645–1655 (2018)
24. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)
25. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
26. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
27. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
28. Moreno-Torres, J.G., Raeder, T., Alaiz-RodríGuez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012)
29. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Advances in Neural Information Processing Systems Workshop (2011)
30. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European Conference on Computer Vision. pp. 102–118. Springer (2016)
31. Roy, S., Siarohin, A., Sangineto, E., Bulo, S.R., Sebe, N., Ricci, E.: Unsupervised domain adaptation using feature-whitening and consensus loss. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9471–9480 (2019)
32. Russo, P., Carlucci, F.M., Tommasi, T., Caputo, B.: From source to target and back: Symmetric bi-directional adaptive gan. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8099–8108 (2018)
33. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Adversarial dropout regularization. In: The International Conference on Learning Representations (2018)
34. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3723–3732 (2018)
35. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceed-

ings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2107–2116 (2017)
36. Shu, R., Bui, H., Narui, H., Ermon, S.: A DIRT-T approach to unsupervised domain adaptation. In: International Conference on Learning Representations (2018)
37. Software, P.: Poser pro 2014. https://www.renderosity.com/mod/bcs/poser-pro-2014/102000, (accessed on 10th-Nov-2019)
38. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2017)
39. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2017)
40. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)
41. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (2018)
42. Zhang, K., Schölkopf, B., Muandet, K., Wang, Z.: Domain adaptation under target and conditional shift. In: International Conference on Machine Learning. pp. 819–827 (2013)
43. Zhang, X., Wong, Y., Kankanhalli, M.S., Geng, W.: Unsupervised domain adaptation for 3D human pose estimation. In: Proceedings of the 27th ACM International Conference on Multimedia (2019)
44. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: The IEEE International Conference on Computer Vision (2017)