Generatively Inferential Co-Training for Unsupervised Domain Adaptation

Can Qin, Lichen Wang, Yulun Zhang, Yun Fu
Northeastern University, USA
qin.ca@husky.neu.com, wanglichenxj@gmail.com,
yulun100@gmail.com, yunfu@ece.neu.edu

Abstract

Deep Neural Networks (DNNs) have greatly boosted the performance on a wide range of computer vision and machine learning tasks. Despite such achievements, DNN is hungry for enormous high-quality (HQ) training data, which are expensive and time-consuming to collect. To tackle this challenge, domain adaptation (DA) could help learning a model by leveraging the knowledge of low-quality (LQ) data (i.e., source domain), while generalizing well on label-scarce HQ data (i.e., target domain). However, existing methods have two problems. First, they mainly focus on the high-level feature alignment while neglecting low-level mismatch. Second, there exists a class-conditional distribution shift even features being well aligned. To solve these problems, we propose a novel Generatively Inferential Co-Training (GICT) framework for Unsupervised Domain Adaptation (UDA). GICT is based on cross-domain feature generation and a specifically designed co-training strategy. Feature generation adapts the representation at low level by translating images across domains. Co-training is employed to bridge conditional distribution shift by assigning high-confident pseudo labels on target domain inferred from two distinct classifiers. Extensive experiments on multiple tasks including image classification and semantic segmentation demonstrate the effectiveness of GICT approach[1].

1. Introduction

In recent years, enormous amounts of images and videos generated online require the help of intelligent methods to analyze their content for downstream exploitation. The advent of Deep Neural Network (DNN) has shown its great capacity in representation learning for vision understanding such as image classification, object detection and semantic segmentation [34, 12, 6, 28]. In spite of its impressive success, DNN requires a large amount of high-quality (HQ) training data which is expensive and time-consuming to collect (*e.g.*, each Cityscapes [6] image takes about 90 minutes to annotate on average).

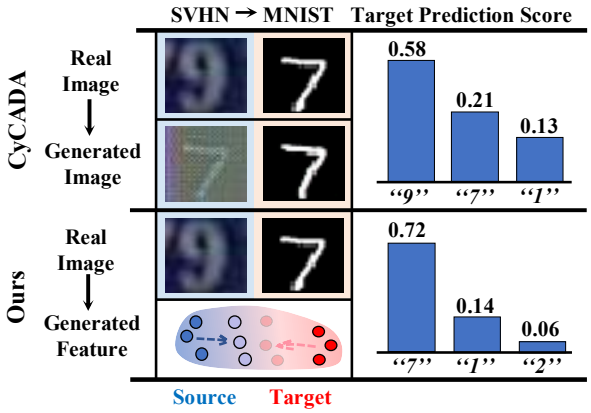

Figure 1. Comparison between CyCADA [13] and ours. CyCADA can mistakenly change the content of source image, *i.e.*, 9 → 7, through generation. Our approach has applied generated features relying on a cycleGAN [47] which draws distant features closer to make them easier for alignment.

To this end, Domain adaptation (DA) is emerged to solve this problem by adapting the model trained on label-abundant low-quality (LQ) data (*i.e.*, source domain) to label-scarce HQ data (*i.e.*, target domain). For instance, synthetic-based DA approaches [6, 13] are attractive which builds a model utilizing the knowledge of noisy and low-resolution synthetic data and generalizes well on the HQ real-scene datasets. However, a model trained on one domain usually performs poorly on other domains since the difference of their characteristics. Even the slight disturbance of training data can severely degenerate its performance [39]. Moreover, the Unsupervised Domain Adaption (UDA) referred as no labels accessible in target domain is the most challenging scenario and has attracted wide attention recently due to its great potential.

The major goal of UDA is to utilize unlabeled samples from the target domain to achieve DA. UDA methods can be summarized in two categories: 1) instance reweighting by estimating the ratio of cross-domain distribu-

[1]Code is available on: https://github.com/ChinTsan01/Generatively-Inferential-Co-Training-for-UDA

tions [14, 13] and 2) feature alignment [9, 14, 39, 5]. Compared to instance re-weighting methods, feature alignment approaches have demonstrated their superiority, especially incorporated with DNN models. Most of feature alignment methods apply a DNN to map raw images into a semantic feature space and mix cross-domain features by minimizing their distance. Currently, generate adversarial networks (GANs) [11] based UDA approach, such as ADDA [39], are employed to align the distributions of two domains relying on a domain classifier (discriminator) and a generator which designs to fool the discriminator. The domain alignment is completed when the discriminator cannot differentiate source and target features.

However, there are two problems in existing UDA methods. Firstly, most methods focus on high-level feature adaptation while neglecting low-level feature structure/information which are crucial for differentiating certain trivial patterns. CyCADA [13] applies a cycleGAN [47] to transform a source image into the "target style" one to reweight input instance. However, as shown in Figure 1, the image content can be mistakenly revised or blurred to degenerate performance of UDA model. Secondly, it is hard to precisely match class-wise conditional distribution without the access to target domain labels.

In this paper, we proposed a novel Generatively Inferential Co-Training (GICT) framework for UDA. To align low-level features, we design the model which extracts the cross-domain features instead of images which are more robust to noise and low resolution. The generated features act as the supplement of raw images which effectively separate the content-irrelevant information (*e.g.*, resolution, deformation, and revision) to effectively bridge the two domains. To correctly draw ambiguous features away from decision boundary, we specifically designed a co-training strategy. It infers highly confident pseudo labels of target domain samples by breaking the closeness of the source set. Instead of inferring pseudo labels from single view, we design a two-classifier strategy to enforce their discrepancy on target domain for inferring highly confident pseudo labels from two distinct views. Furthermore, simply average the prediction results from the two classifiers ignores the difference information. Thus, we proposed an label graph network to further explore the prediction accuracy between two classifiers and across each pair of labels. Moreover, a channel attention layer [46] followed by the concatenation layer is further adapt to align high-level real and generated image features. In summary, the contributions of our framework are below:

- We proposed a feature translation framework which incorporates "source content" and "target style" as the supplement of raw data for low-level domain adaption.
- We designed a novel co-training strategy to draw the ambiguous features to their corresponding side by inferring pseudo labels with high confidence relying on two distinct classifiers and inference procedure.
- A label correlation graph network is further proposed to explore the label-label relations between the two classifiers and achieve higher classification accuracy.

2. Related Works

Our proposed approach is mainly related to unsupervised domain adaptation techniques and co-training mechanism.

2.1. Unsupervised Domain Adaptation (UDA)

Over the past few years, UDA has attracted increasing attention to reduce the annotation cost. The key challenge of UDA is that the distribution shift exists between the source domain $P(s)$ and the target domain $P(t)$ where $P(s) \neq P(t)$. It violates the assumption of conventional machine learning methods that training and test samples share the same distribution. To mitigate domain shift, many approaches have been proposed [9, 14, 5, 40, 42, 8] and they can be summarized into two lines: 1) instance re-weighting and 2) feature alignment. Instance re-weighting methods attempt to assign the weights of the training data to adapt the distribution of the target domain $P(t)$ based on the estimated ratio $P(s)/P(t)$ of two-domain distributions [14]. Feature alignment methods address this problem by learning a mapping function $f(\cdot)$ to map the raw images into a latent feature space where the representations of two domains can be aligned by minimizing their distance. A typical subspace learning approach to this problem is to map both the source and the target samples into a shared subspace based on the metric learning [26, 41] and dictionary learning techniques [37].

Inspired by the impressive performance of deep learning in visual recognition [16, 34, 12], many deep learning-based domain adaptation methods have been proposed. A natural idea is to minimize certain kinds of divergence or distance measured by first-order or second-order statistics between deep features across domains. Various methods, such as Maximum Mean Discrepancy (MMD) [22] or Deep Correlation Alignment (CORAL) [35] have been proposed.

Other popular approaches utilize adversarial learning training to learn domain invariant representations from generator by fooling a domain classifier (discriminator) with the help of gradient reverse [9] or GAN [39]) (ADDA) until the discriminator is unable to distinguish the features between two domains. [13] extends ADDA by introducing an cycleGAN-based [47] instance re-weighting approach. It transforms source images to "target style", which effectively transfer low-level visual feature to target domain. However, since the limitation of decoder, the generated images suffer content revision, deformation, and low-resolution. To this end, we proposed to translate in-

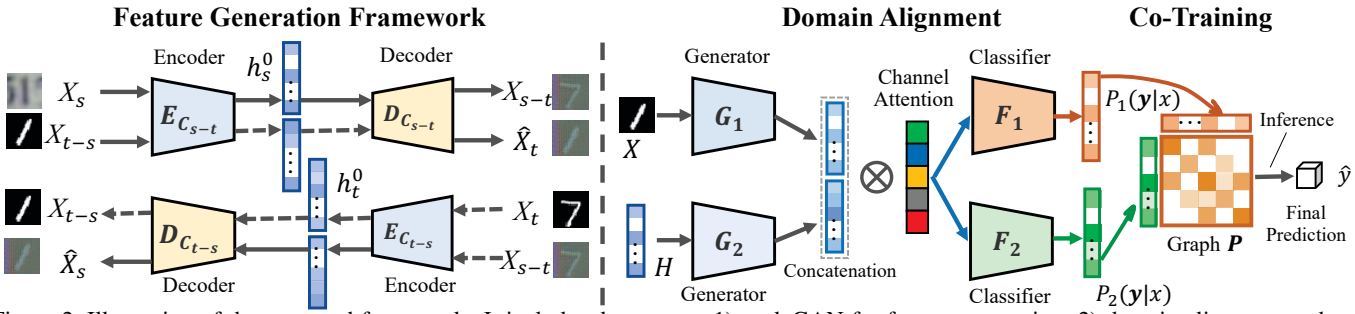

Figure 2. Illustration of the proposed framework. It includes three parts: 1) cycleGAN for feature generation; 2) domain alignment and 3) co-training. $\otimes$ denotes element-wise product. Ec and Dc represent encoder and decoder in cycleGAN respectively. $\hat{X}$ indicates reconstructed image of X and its footnote, *i.e.*, st or ts, denotes the direction of transformation.

termediate features to avoid above issues, while low-level visual knowledge is still preserved.

2.2. Co-training

Co-training is one of the most typical and well known semi-supervised learning approaches [3]. It trains multiple classifiers on distinct views, and assigns the pseudo labels of the unlabeled instances iteratively. The reasonability of co-training lies in the assumption that the false pseudo labels can be corrected in an iterative way. Therefore, the main issue of co-training is pseudo labels as well as its confidence evaluation. Many theoretical investigations prove that the upper bound of co-training methods is largely influenced by the diversity of multiple classifiers [3] for which we pay a lot of attention.

Co-training has been applied to domain adaptation tasks [48, 7] as both semi-supervised learning and domain adaptation have partially unlabeled data for training. [48, 44, 43] regards the target labels as hidden variables that can be learned by adapting an easy-to-hard strategy to select the "high confident" pseudo labels gradually based on the softmax scores and the regulation of spatial prior knowledge. However, their evaluation is based on the result of one classifier (*i.e.* view) which is subjective. [7] applies graph models to infer the soft pseudo labels of target domain. However, this graph relies on shadow models which limits its capacity in representation learning [1, 36].

3. Proposed Method

As shown in Figure 2, our model consists of three parts: 1) feature augmentation, 2) domain alignment, and 3) co-training. In general, generated features serve as the supplements for raw images to bridge two domains at low-level, and co-training breaks the closeness of source set and encourages class-wise alignment. All the experiments in tables have been repeated 3 times and we report the average one. We will analyze each component in detail and explain the training procedure in the following sections.

3.1. Feature Augmentation

To mitigate the low-level domain shift, we introduce the generated features which mix the "content" of the source domain and the "style" of the target domain as the supplements for original features drawn from $\{X_s, X_t\}$. The generated features are based on translating images across domains.

The feature generation procedure involves the mapping from the source to target $G_{S\to T}(\cdot)$, the inverse mapping $G_{T\to S}(\cdot)$, and the discriminators $D_S(\cdot)$ and $D_T(\cdot)$ for each domain. The generators $G_{S\to T}(\cdot)$ and $G_{S\to T}(\cdot)$ attempt to fool $D_T(\cdot)$ and $D_S(\cdot)$ respectively, and the $D_T(\cdot)$, $D_S(\cdot)$ are employed to classify whether the generated image is real or not. This is accomplished by achieving the following objects:

$$\begin{aligned} L^1_{adv}(G_{S\to T}, D_T) =& \mathbb{E}_{x_t\sim X_t}[\log D_T(x_t)] \\ &+\mathbb{E}_{x_s\sim X_s}[\log 1 - G_{S\to T}(x_s)], \end{aligned} \tag{1}$$

$$\begin{aligned} L^2_{adv}(G_{T\to S}, D_S) =& \mathbb{E}_{x_s\sim X_s}[\log D_S(x_s)] \\ &+\mathbb{E}_{x_t\sim X_t}[\log 1 - G_{T\to S}(x_t)]. \end{aligned} \tag{2}$$

However, the adversarial training process is unstable and prone to failure. To this end, we apply a cycle-consistency loss to enforce the consistency between the input real and reconstructed images as shown below:

$$\begin{aligned} &L_{cyc}(G_{S\to T}, G_{T\to S}) \\ &= \mathbb{E}_{x_s\sim X_s}[\|G_{T\to S}((G_{S\to T}(x_s))) - x_s\|_1] \\ &\quad + \mathbb{E}_{x_t\sim X_t}[\|G_{S\to T}((G_{T\to S}(x_t))) - x_t\|_1]. \end{aligned} \tag{3}$$

The generator $G_{S\to T}(\cdot)$ and $G_{T\to S}(\cdot)$ have learned mappings across different domains. The representations learned by generators involve mutual characteristics of two domains. Therefore, we apply them to extract their inside features $\mathbf{h}^0_s$ and $\mathbf{h}^0_t$ for feature augmentation:

$$\mathbf{h}^0_s = G_{S\to T}(x_s|\Theta^l_{S\to T}), \tag{4}$$

$$\mathbf{h}^0_t = G_{T\to S}(x_t|\Theta^l_{T\to S}), \tag{5}$$

where Θ^l denotes the parameters before the l-th layer of the generative network.

3.2. Domain Alignment

Suppose we have the access to a labeled source image x_s and its corresponding label y_s, drawn from the set of source images $\{X_s, Y_s\}$. We also have the unlabeled target image x_t drawn from target image set X_t. The goal of UDA is to build a model that generalizes well on target domain by transferring the knowledge from source domain. Such two domains belong to different marginal distributions, *i.e.*, $P(X_s) \neq P(X_t)$, as well as distinct conditional distributions, *i.e.*, $P(y_s|X_s) \neq P(y_t|X_t)$. Then, the models trained only by using source samples perform poorly on target domain.

Inspired by MDA [32], our proposed method aligns the distributions of two domains with two feature generator networks $G_1(\cdot)$, $G_2(\cdot)$ and two classifier networks $F_1(\cdot)$ and $F_2(\cdot)$. The generator $G_1(\cdot)$ is utilized to extract the feature $\mathbf{h}_i^1 \in \mathbb{R}^d$ of the i-th input image x_i from the input set $\{X_s, X_t\}$ as:

$$\mathbf{h}_i^1 = G_1\left(x_i|\Theta_g^1\right), \tag{6}$$

where $G_1(\cdot)$ is the generative function, parameterized by Θ_g^1.

The features $\{\mathbf{h}_s^0, \mathbf{h}_t^0\}$ obtained in Eqs. (4) and (5) are fed into the second generative network $G_2(\cdot)$ for domain alignment and concatenated with the features $\mathbf{h}^1$ obtained in Eq. (6). The concatenated features are denoted as $\mathbf{h}_t = \begin{bmatrix} G_2(\mathbf{h}_t^0) \\ \mathbf{h}_t^1 \end{bmatrix}$ and $\mathbf{h}_s = \begin{bmatrix} G_2(\mathbf{h}_s^0) \\ \mathbf{h}_s^1 \end{bmatrix}$. Since features $\mathbf{h}_s, \mathbf{h}_t \in \mathbb{R}^{H\times W\times C}$ come from different domains, it is necessary to weight the importance of each for the selection of useful ones. The attention mechanism can help to explore the channel-wise dependence among the features of two domains for concatenating them smoothly in the feature space. Therefore, we apply a channel-wise attention [46] to accomplish this object:

$$z_c = \frac{1}{H\times W}\cdot\sum_i^H\sum_j^W h_c(i,j), \tag{7}$$

$$\hat{h}_c = \varphi(W_U\delta(W_D z_c))\cdot h_c, \tag{8}$$

where $h_c \in \mathbb{R}^{H\times W}$ is the slice of concatenated feature $\mathbf{h}$ at the c-th channel, and H and W represent the height and width of h respectively. $\varphi(\cdot)$ and $\delta(\cdot)$ denote the sigmoid gating and ReLU function respectively. W_D is the weight set of a convolutional layer, which acts as channel-downscaling with reduction ratio r. After being activated by ReLU, the low-dimension signal is then increased to $\hat{h}_c$ with ratio r by a channel-upscaling layer, whose weight set is W_U where $W_U \bigcup W_D = \mathcal{W}$.

The two classifiers $F_1(\cdot)$ and $F_2(\cdot)$ take the features $\hat{\boldsymbol{h}}_i \in \{\hat{H}_s, \hat{H}_t\}$ from generators $G_1(\cdot)$ and $G_2(\cdot)$ as inputs and classify them into K classes:

$$p_1(\mathbf{y_i}|x_i) = F_1\left(\hat{\boldsymbol{h}}_i|\Theta_f^1\right), \tag{9}$$

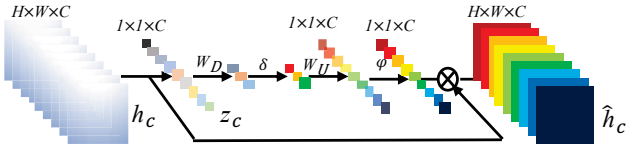

Figure 3. The illustration of channel-wise attention (CA). $\bigotimes$ denotes element-wise product. $H \times W \times C$ represent the size of feature $\mathbf{h}$ with height H, width W and C channels.

$$p_2(\mathbf{y_i}|x_i) = F_2\left(\hat{\boldsymbol{h}}_i|\Theta_f^2\right), \tag{10}$$

where $p_1(\mathbf{y_i}|x_i)$ and $p_2(\mathbf{y_i}|x_i)$ denote the K-dimensional probabilistic softmax results of $F_1(\cdot)$ and $F_2(\cdot)$ for input x. Θ_f^1 and Θ_f^2 are parameters of $F_1(\cdot)$ and $F_2(\cdot)$ respectively.

To train the model, the total loss consists of two parts: task loss and discrepancy loss. Similar as most UDA methods, the object of task loss is to minimize the empirical risk on source domain $\{X_s, Y_s\}$, which is formulated as follows:

$$\begin{aligned} L_{cls}(X_s, Y_s) = &- \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} \\ &\sum_{k=1}^{K} 1_{[k=y_s]}\log(p((\mathbf{y}=y_s)|G(x_s|\Theta_g))), \end{aligned} \tag{11}$$

where $x_{s,i} \in \boldsymbol{X_s}$, $y_s \in \{1, ..., K\}$ and K indicates total quantity of classes. The discrepancy loss is calculated as the l_1 distance between the softmax scores of two classifiers on target domain X_t:

$$L_{dis}(X_t) = \mathbb{E}_{x_t\sim X_t}[|p_1(\mathbf{y}|x_t) - p_2(\mathbf{y}|x_t)|]. \tag{12}$$

The details of training procedure is described in Section 3.4.

3.3. Co-training

Although MDA has aligned the features of two domains into a common space, the ambiguous features near the decision boundary can be classified into the wrong side due to the poor initialization of generator and bias on source domain. In order to break the closeness amo ng source labels and correct the wrongly classified target samples, we propose a co-training mechanism to infer highly confident pseudo labels to further fine-tune the networks. The key issue of co-training lies in the build of multiple and diversified classifiers to investigate target samples from distinct views. As two classifiers are provided in Section 3.2, it is natural to infer pseudo labels of target samples based on these two classifiers.

The co-training procedure consists of two parts: 1) label-level correlation fine tuning and 2) pseudo label inference. The prediction results from either classifier $F_1(\cdot)$ or $F_2(\cdot)$ could be considered as the final classification results. Averaging the predictions is also an effective and efficient strategy. However, the trivial prediction differences between $F_1(\cdot)$ and $F_2(\cdot)$ could still existing critical information of the classification boundaries. To this end, we expect to fully

Algorithm 1 Intermediate Feature Generation

Input: Source image set X_s, target image set X_t. The number of training epochs T. The randomly initialized generators ($G^0_{T\to S}$, $G^0_{S\to T}$) and discriminators (D^0_T, D^0_S).

$t \leftarrow 0$
while $t < T$ **do**
  $t \leftarrow t + 1$.
  update D^{t-1}_T to D^t_T by Eq. (1).
  update D^{t-1}_S to D^t_S by Eq. (2).
  update ($G^{t-1}_{T\to S}$, $G^{t-1}_{S\to T}$) to ($G^t_{T\to S}$, $G^t_{S\to T}$) by Eqs. (1), (2), (3).
end while
Extract features h^0_s by Eq. (4) and h^0_t by Eq. (5).
return Feature set $\{h^0_s, h^0_t\}$.

utilize the classification results across the two classifiers to further boost the learning performance.

We first multiply the output vectors $\mathbf{p}_1(\mathbf{y_i}|x_i)$ and $\mathbf{p}_2(\mathbf{y_i}|x_i)$ of classifiers $F_1(\cdot)$ and $F_2(\cdot)$ to achieve the matrix $\mathbf{P} \in \mathbb{R}^{K\times K}$ containing the joint classification knowledge of two classifiers.

$$\mathbf{P}_i = \mathbf{p}_1(\mathbf{y_i}|x_i) \cdot \mathbf{p}_2(\mathbf{y_i}|x_i)^\top, \tag{13}$$

where $\mathbf{P}_i$ can be considered as the dot-similarity of each pair of labels across the two classifiers. After that, the largest element v on the trace of $\mathbf{P}_i$ is selected as the prediction:

$$\hat{v}_i, \hat{y}_i = \arg\max_{v_i, y_i} \mathrm{Tr}(\mathbf{P_i}), \tag{14}$$

where $\hat{v}_i \in \hat{V}$, and $\hat{y}_i \in \hat{Y}$ indicate its index in the trace as well as the final inference result. By this way, the model further considers the prediction scores between $F_1(\cdot)$ and $F_2(\cdot)$ in label-label which further explores information residing inside the trivial classification differences.

All the test samples are processed in this way to achieve the final output results and we select the $\mathcal{R}$ ratio of samples in $\hat{Y}_t \subset \hat{Y}$ on training set as pseudo labels $\hat{y}^*_t \in \hat{Y}^*_t$. The ratio $\mathcal{R}$, starting from 0, keeps increasing linearly through the optimization of the model. All the pseudo labels are applied to fine-tune the model as follows:

$$L_{co}(\boldsymbol{X}_t, \hat{Y}^*_t) = -\mathbb{E}_{(\mathbf{x}_t, \hat{y}^*_t)\sim(X_t, \hat{Y}^*_t)} \sum_{k=1}^{K} 1_{[k=y_t]} \log(p((\mathbf{y} = \hat{y}^*_t)|G(\mathbf{x}_t|\boldsymbol{\Theta}_g))). \tag{15}$$

3.4. Training Procedure

Let's sum up the discussions in previous sections into the whole optimization process, which consists of initialization and 3 steps in total:

Initialization. Firstly, given the images of two domains $\{X_s, X_t\}$, we initialize the intermediate features $\{h^*_s, h^*_t\}$ by optimizing the sum of losses obtained in Eq. (6), Eq. (7), and Eq. (8):

$$\{\mathbf{h}^*_s, \mathbf{h}^*_t\} = \arg\min_{G_{TS}, G_{ST}, D_S, D_T} L^1_{adv} + L^2_{adv} + \beta L_{cyc}, \tag{16}$$

where β is the loss weight and assigned as 0.1 in the model.

Algorithm 2 The proposed domain alignment algorithm

Input: Labeled source set $\{X_s, Y_s\}$, target image set X_t. The number of training epochs T. The randomly initialized generators (G^0_1, G^0_2), classifiers (F^0_1, F^0_2) and weight matrix $\mathcal{W}^0$.

$t \leftarrow 0$
while $t < T$ **do**
  $t \leftarrow t + 2$.
  update (F^{t-2}_1, F^{t-2}_2) to (F^{t-1}_1, F^{t-1}_2) by **Step1**.
  update (G^{t-2}_1, G^{t-2}_2,$\mathcal{W}^{t-2}$) to (G^{t-1}_1, G^{t-1}_2, $\mathcal{W}^{t-1}$) by **Step1**
  update (F^{t-1}_1, F^{t-1}_2) to (F^t_1, F^t_2) by **Step2**.
  update (G^{t-1}_1, G^{t-1}_2,$\mathcal{W}^{t-1}$) to (G^t_1, G^t_2),$\mathcal{W}^t$) by **Step3**.
end while
Infer target set labels $\hat{Y}_t$ by Eqs. (13) and (14).
return Inference results $\hat{Y}_t$.

Step1. In this step, we introduce image-label pairs $\{X_s, Y_s\}$ on the source domain to train both two classifiers and two generators, which makes them learn discriminative features and clear decision boundaries on source domain. The objects are accomplished by minimizing the function:

$$\min_{G_1, G_2, \mathcal{W}, F_1, F_2} L_{cls}(X_s, Y_s), \tag{17}$$

and we apply the model trained on source domain to infer the highly confident pseudo labels on target set for later co-training according to Eq. (13) and Eq. (14):

$$\hat{v}, \hat{y}_t = \arg\max_{v,y} \mathrm{Tr}(\mathbf{P}), \tag{18}$$

and select the $\mathcal{R}\cdot\|\hat{Y}_t\|$ number of inferred labels as the high-confident pseudo labels $\hat{Y}^*_t$ for further fine-tuning.

Step2. It is required to train two classifiers $F_1(\cdot)$ and $F_2(\cdot)$ with the discrepancy loss L_{dis} in Eq. (12), co-training loss L_{co} in (15), and source loss L_{cls} with the fix generators. The discrepancy loss, which requires to be maximized, helps classifiers detect target samples beyond the support of the source and co-training loss. It further accelerates this process by adjusting its decision boundary towards highly confident target samples. The source loss is applied to avoid the departure of decision boundary on the source through its adjustment. The objective function is follows:

$$\min_{F_1, F_2} L_{cls}(X_s, Y_s) + L_{co}(X_t, \hat{Y}^*_t) - \alpha L_{dis}(X_t), \tag{19}$$

where α is the loss weight and assigned as 0.1 in the model.

Step3. In this step, we train the generators $G_1(\cdot)$ and $G_2(\cdot)$ to minimize the discrepancy and co-training losses. It is crucial for aligning domains, as the two losses help to draw the features $\hat{\mathbf{h}}_t^*$ on target domain towards the neighbouring source ones given the support of source learned in **Step3**. The objective function is formulated as:

$$\min_{G_1,G_2,\mathcal{W}} L_{co}(X_t, \hat{Y}_t^*) + \alpha L_{dis}(X_t). \tag{20}$$

The optimization process of the whole framework begins with the **Initialization** to generate features as the inputs for generator $G_2(\cdot)$. The loop between **Step1**, **Step2**, and **Step3** starts after that and stops when reaching a certain quantity of steps assigned manually.

4. Experiments

In this section, we provide comprehensive evaluations of the proposed method on the tasks of digits recognition, image classification and semantic segmentation and compare with the state-of-the-art approaches on these tasks. Details of experiments are described in following sections.

4.1. Digits Recognition

In the first experiment, we evaluate the proposed method on several popular digit datasets with different characteristics. Following [32], we apply the same architecture of generator $G_1(\cdot)$, and two classifiers $F_1(\cdot)$ and $F_2(\cdot)$. The architecture of $G_2(\cdot)$ is based on the revision of VGG-16. We apply momentum Stochastic Gradient Descent (SGD) as the optimizer implemented on the platform of PyTorch [2]. The learning rate is 0.0002 and momentum is 0.9 with weight decay 0.0005. All models have been trained for 100 epochs in the batch size 128. The training procedure of CycleGAN follows the protocol of CyCADA [13] with the batch size 100, epoch 100, learning rate 0.001, and optimization method Adam. The architecture is based on LeNet and there are six residual layers between the encoder and the decoder. We extract features for augmentation from the first fully connection layer. The influence of feature extraction layer will be analyzed in Section 4.4.

We organize four types of adaptation scenarios given the access to four digits datasets including MNIST [17], USPS [31], Street View House Numbers (SVHN) [25], and Synthetic Number (SynNUM) [9]. Here are the details of each adaptation scenario:

MNIST $\leftrightarrow$ USPS. In this scenario, both MNIST and USPS consist of white digits ranging from 0 to 10 on the solid black background. In general, this is regarded as one of the easiest adaptation scenario since many similarities exist between two datasets. We evaluate the proposed method following the first protocol (P1) on [32], where the MNIST test set is composed of 2,000 images and USPS test set contains 1,800 images. All images of USPS have been re-sized into 28×28 pixels to fit the size of images in MNIST.

[2] https://pytorch.org/

Table 1. Quantitative results (%) on Digits Datasets.

Method	SVHN $\downarrow$ MNIST	USPS $\downarrow$ MNIST	MNIST $\downarrow$ USPS	SynNum $\downarrow$ SVHN
MMD [21]	64.8	73.5	88.5	-
DANN [10]	71.1	73.0	77.1	91.1
DSN [4]	82.7	-	91.3	-
ADDA [39]	76.0	90.1	89.4	-
CoGAN [20]	-	89.1	91.2	-
UNIT [19]	90.5	93.5	95.9	-
CyCADA [13]	90.4	96.5	95.6	-
MDA [32]	96.2	94.1	94.2	89.9
DeepJDOT [2]	96.7	96.4	95.7	
DIRT-T [33]	96.7	99.4	95.7	
Ours (Feat)	96.7	94.8	94.7	91.9
Ours (Co)	97.1	95.2	94.9	92.3
Ours (Feat+Co)	**98.7**	**96.6**	**96.2**	**93.2**

SVHN $\rightarrow$ MNIST. The Street View House Numbers (SVHN) dataset contains digits images, which are captured on real scenes and cropped into the size of 32×32. These two datasets belong to two distinct distributions, because SVHN images are more diverse with colorful and clustered background while MNIST images are simply black and white. It is expected that DNN is able to learn rich knowledge in SVHN, which can cover the domain of MNIST.

SynNum $\rightarrow$ SVHN. The Synthetic Number (SynNum) dataset, which consists of about 500,000 images, is collected from Windows^{TM} fonts by varying the text, positioning, orientation, background, stroke colors, and the amount of blur. The variations were chosen manually to simulate those of SVHN while they still share distinctions. The biggest difference is the structured clutter in the background of SVHN images.

Result Analyses. The quantitative results and comparison on digits datasets are summarized in Table 1. The proposed methods outperform the directly comparable methods CyCADA [13] and MDA [32] on all adaptation scenarios and other state-of-the-art ones. Although one of the largest domain gaps appears on SVHN-to-MNIST, ours exhibit the largest superiority. However, on the domain pair USPS-MNIST with the least domain gap, our method slightly outperform previous methods, which indicates that our proposed methods are better at dealing with the scenarios with larger domain shift. This phenomenon can be explained by the facts that the features between significantly different domain pairs distribute more randomly than the slightly different ones. Our proposed method leaves more potential to correct the mistakenly classified target samples given better support of the source samples and highly confident pseudo

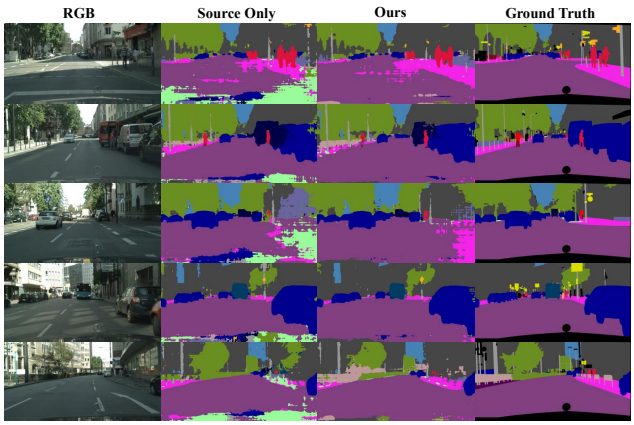

Figure 4. Example semantic segmentation results in the GTA5-to-Cityscapes adaptation scenario.

labels in the target set.

The last three rows of Table 1 investigate the ablation studies between feature augmentation, co-training, and the full version of proposed method (*i.e.*, feature augmentation + co-training). The superiority of our full version method over single version methods (*i.e.,* "Feat" and "Co") proves that the combination of feature augmentation and co-training can improve the performance mutually. In comparison of co-training and feature augmentation, the former sightly outperforms feature augmentation on all adaptation scenarios, which means that breaking the closeness of the source set is crucial for aligning the cross-domain features.

In order to better understand the distribution of features, we employ the visualization technique to analyze generator features using t-sne algorithm [23] which are shown in Figure 5. In comparison of (b) and (c), even though some target features not tightly mixed with source features, our features are more clustered which indicates that pseudo labels are helpful in drawing ambiguous features towards the corresponding side. The other advantage comes from the enlarged gap which makes the features more separable and decision boundary more robust.

4.2. Image Classification on VisDA-2017 Dataset

To further evaluate our proposed method, we conduct experiments on image classification tasks and compare it with the state-of-the-art methods. VisDA-2017 Dataset [27] is applied to evaluate the synthetic-to-reality adaptation scenario which is composed of the synthetic-object images generated by 3-D CAD models for training and objects collected from MS-COCO [18] for validation as well as those from YouTube BoundingBoxes [29] for testing. The VisDA-2017 dataset consists of 280,000 images in total covering 12 classes.

Implementation Details. The architecture of our generator network and classifiers networks follows those of [32] with ResNet-101 [12] as the backbone for fair comparison. We apply Adam [15] as the optimizer with learning rate

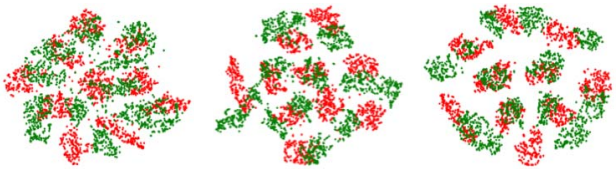
(a) (b) (c)

Figure 5. The t-sne [23] visualization results of features on MNIST-to-USPS obtained by (a) Source Only, (b) MDA [32] and (c) Ours. The feature points of source and target domains are indicated by red and green spots respectively.

0.001. Batch size is set to be 32 and we report the results achieved on validation set after 20 epochs. As for feature generation, we apply the VGG-16 [34] except fully connected layers for encoder and de-conv network [24] as the decoder with 9 residual blocks [12] between the encoder and decoder. The features applied for augmentation are extracted from the last residual layer which incorporates the mutual information across two domains.

Results Analyses. The quantitative results on VisDA-2017 dataset are summarized in Table 2 where our method on average outperforms the baseline methods (MDA, DANN and DeepJDOT) by a large margin. Although not achieving the best results on some of objects, ours is very close to the state-of-the-art methods and has achieved much improvement on the "skateboard" object which is challenging for recognizing. In addition, our proposed method outperforms the "Source Only" method on all categories which demonstrates the superiority of our adaptation in overcoming the negative transfer problem.

4.3. Semantic Segmentation

As it is the heavy work to manually label each pixel of whole image, it is necessary and urgent to propose annotation efficient methods on the semantic segmentation of which UDA is a promising solution. In this section, we further conduct experiments on domain adaptation of semantic segmentation task which involves a pixel-wise adaptation scenario required to bridge the domain shift on every pixel.

Datasets. To evaluate our proposed method, we applied two benchmarks including synthetic GTA5 [30] dataset and real-scene Cityscapes [6] dataset of which both focus on street scenes segmentation. GTA5 dataset consists of 24,966 images collected from the game world of Grand Theft Auto V. Although GTA5 images simulates well in illumination, texture and colors, there still exists a great domain shift caused by noise, deformed objects and different resolution from the real street scenes. The real world Cityscapes dataset is composed of 2975 urban street images for training, 500 images for validation as well as 1525 testing images. Both GTA5 and Cityscapes dataset share the same set of 19 categories which is straightforward for evaluation. As most of semantic segmentation methods, we use

Table 2. Quantitative image classification results (%) on VisDA-2017 Dataset.

	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbd	train	truck	MEAN
Source Only	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
MMD [21]	87.1	63.0	76.5	42.0	**90.3**	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
DANN [10]	81.9	**77.7**	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
MDA [32]	87.0	60.9	**83.7**	64.0	88.9	**79.6**	84.7	**76.9**	88.6	40.3	83.0	25.8	71.9
DeepJDOT [2]	85.4	50.4	77.3	**87.3**	69.1	14.1	**91.5**	53.3	**91.9**	31.2	**88.5**	**61.8**	66.9
Ours	**87.6**	60.6	81.6	72.1	87.8	62.9	89.7	68.5	88.8	**76.1**	83.2	20.0	**73.1**

Table 3. Quantitative semantic segmentation results (%) on GTA5 to Cityscapes Dataset.

	road	sdwk	bldng	wall	fence	pole	light	sign	vgttn	trrn	sky	person	rider	car	truck	bus	train	mcycl	bcycl	mIOU
Source Only	36.4	14.2	67.4	16.4	12.0	20.1	8.7	0.7	69.8	13.3	56.9	37.0	0.4	53.6	10.6	3.2	0.2	0.9	0.0	22.2
DANN [10]	64.3	23.2	73.4	11.3	18.6	29.0	31.8	14.9	82.0	16.8	73.2	53.9	12.4	53.3	20.4	11.0	5.0	18.7	9.8	32.8
CyCADA [13]	79.1	33.1	77.9	23.4	17.3	**32.1**	33.3	**31.8**	81.5	26.7	69.0	**62.8**	14.7	74.5	20.9	25.6	**6.9**	18.8	20.4	39.5
MDA [32]	**90.3**	31.0	78.5	19.7	17.3	28.6	30.9	16.1	**83.7**	30.0	69.1	58.5	19.6	**81.5**	23.8	30.0	5.7	25.7	14.3	39.7
AdaSegNet [38]	86.5	36.0	**79.9**	**23.4**	23.3	23.9	**35.2**	14.8	83.4	33.3	**75.6**	58.5	**27.6**	73.7	**32.5**	**35.4**	3.9	30.1	**28.1**	42.4
Ours	88.6	**41.3**	76.4	23.3	**26.1**	24.3	32.8	23.1	82.3	**37.4**	73.3	62.2	24.8	73.3	29.6	33.9	4.6	**33.4**	24.3	**42.8**

mIOU as the evaluation metric [13].

Implementation Details. In this part, we apply DRN-D-105 [45] as the backbone for the generator G_1, and G_2 is based on the revision of VGG-16. Following [32], We apply Momentum SGD as the optimization method with learning rate 0.001 and momentum rate 0.9. Due to the limitation of GPU memory, the batch size is set to be 1 and we report the results after 50,000 iterations. We employ the same architecture of cycleGAN as Section 4.2 and the augmented features are extracted from the last residual layer.

Results Analyses. The results of the evaluations are shown in Table 3. Compared with the baseline methods, *i.e.* MDA and CyCADA, our proposed method boost the result of mIOU to 3 percent approximately which is a significant improvement. The superiority of ours lies in the tiny objects including fence, motorcycle with sharp edges but complicated texture. Example segmentation results are presented in Figure 4. Compared with the model trained only with GTA5 images which is likely to mis-classify road, our segmentation results are more consistent and smooth.

4.4. Analyses

To further evaluate the performance of our proposed method on different conditions, we conduct experiments of sensitive analyses under different feature extraction layers in Figure 6 (a) and convergence analysis of different pseudo-label-selection strategies in Figure 6 (b) on the domain pair of MNIST-to-USPS.

As shown in Figure 6 (a), through the move of feature extraction layers towards the decoder (Layer 1→Layer 6→Decoder) in cycleGAN, the classification accuracy keeps increasing. It can be explained by the facts that the features neighbouring to decoder contain more mutual information of two domains and are more useful for augmenting the original image features.

In Figure 6 (b) "Quadratic" and "Linear" refer to quadratic and linearly increasing $\mathcal{R}$ for the selection of pseudo labels. "Step" refers to a step function where $\mathcal{R}$ rises from 0 to 0.5 at the 20-th epoch. To continuous increase $\mathcal{R}$ is helpful for the selection of highly confident pseudo labels as the representation learned by generators become better through the optimization.

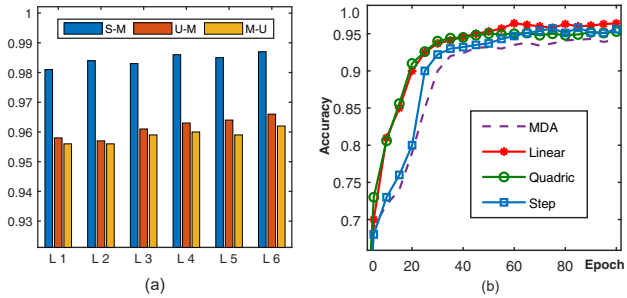

Figure 6. (a) Performance of three adaptation scenario (*i.e.*, SVHN-to-MNIST, USPS-to-MNIST and MNIST-to-USPS) under different augmented feature extraction layer ranging from Layer-1 (L1) to Layer-6 (L6) in 6 residual blocks. (b) Convergence analysis of proposed method on adaptation scenario MNIST-to-USPS.

5. Conclusion

In this paper, we propose a novel Generatively Inferential Co-Training (GICT) framework for Unsupervised Domain Adaptation (UDA). A cross-domain feature generation framework and a co-training strategy are deployed to achieve UDA. The feature generation model aligns the distributions at low level by translating and generating images across the source and target domains. The co-training strategy is further proposed to bridge the class-wise conditional distribution shift by assigning high-confident pseudo labels on target domain samples inferred from a two-classifiers (*i.e.*, views) network structure. Extensive experiments demonstrate the superiority of GICT approach on benchmarks including digits recognition, image classification and semantic segmentation.

Acknowledgments: This research is supported in part by the NSF IIS award 1651902 and U.S. Army Research Office Award W911NF-17-1-0367.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, pages 447–463, 2018.

[3] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NIPS*, pages 343–351, 2016.

[5] Yue Cao, Mingsheng Long, and Jianmin Wang. Unsupervised domain adaptation with distribution matching machines. In *AAAI*, 2018.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[7] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *ECCV*, pages 37–52, 2018.

[8] Jiahua Dong, Yang Cong, Gan Sun, and Dongdong Hou. Semantic-transferable weakly-supervised endoscopic lesions segmentation. In *ICCV*, October 2019.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[14] Mohammad Nazmul Alam Khan and Douglas R Heisterkamp. Adapting instance weights for unsupervised domain adaptation using quadratic mutual information and subspace learning. In *CVPR*, 2016.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

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

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[19] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, pages 700–708, 2017.

[20] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NIPS*, pages 469–477, 2016.

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[22] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[24] Rahul Mohan. Deep deconvolutional networks for scene parsing. *arXiv preprint arXiv:1411.4101*, 2014.

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

[26] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[27] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[28] Can Qin, Maoguo Gong, Yue Wu, Dayong Tian, and Puzhao Zhang. Efficient scene labeling via sparse annotations. In *Workshops at the AAAI*, 2018.

[29] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, pages 5296–5305, 2017.

[30] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, pages 102–118. Springer, 2016.

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[32] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018.

[33] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*. Springer, 2016.

[36] Gan Sun, Yang Cong, Qianqian Wang, Bineng Zhong, and Yun Fu. Representative task self-selection for flexible clustered lifelong learning. *arXiv preprint arXiv:1903.02173*, 2019.

[37] Gan Sun, Yang Cong, and Xiaowei Xu. Active lifelong learning with" watchdog". In *AAAI*, 2018.

[38] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018.

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[40] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *ACM MM*, pages 402–410. ACM, 2018.

[41] Lichen Wang, Zhengming Ding, and Yun Fu. Learning transferable subspace for human motion segmentation. In *AAAI*, 2018.

[42] Lichen Wang, Zhengming Ding, and Yun Fu. Low-rank transfer human motion segmentation. *IEEE Transactions on Image Processing*, 28(2):1023–1034, 2019.

[43] Lichen Wang, Zhengming Ding, Seungju Han, Jae-Joon Han, Changkyu Choi, and Yun Fu. Generative correlation discovery network for multi-label learning. In *ICDM*, 2019.

[44] Lichen Wang, Zhengming Ding, Zhiqiang Tao, Yunyu Liu, and Yun Fu. Generative multi-view human action recognition. In *ICCV*, 2019.

[45] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, pages 472–480, 2017.

[46] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[48] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018.