

Model Adaptation: Unsupervised Domain Adaptation without Source Data

Rui Li¹, Qianfen Jiao¹, Wenming Cao³, Hau-San Wong¹, Si Wu²

¹Department of Computer Science, City University of Hong Kong

²School of Computer Science and Engineering, South China University of Technology

³Department of Statistics and Actuarial Science, The University of Hong Kong

rui1i52-c@my.cityu.edu.hk, qjiao4-c@my.cityu.edu.hk, wmingcao@hku.hk

[✉]cshswong@cityu.edu.hk, cswusi@scut.edu.cn

Abstract

In this paper, we investigate a challenging unsupervised domain adaptation setting — unsupervised model adaptation. We aim to explore how to rely only on unlabeled target data to improve performance of an existing source prediction model on the target domain, since labeled source data may not be available in some real-world scenarios due to data privacy issues. For this purpose, we propose a new framework, which is referred to as collaborative class conditional generative adversarial net to bypass the dependence on the source data. Specifically, the prediction model is to be improved through generated target-style data, which provides more accurate guidance for the generator. As a result, the generator and the prediction model can collaborate with each other without source data. Furthermore, due to the lack of supervision from source data, we propose a weight constraint that encourages similarity to the source model. A clustering-based regularization is also introduced to produce more discriminative features in the target domain. Compared to conventional domain adaptation methods, our model achieves superior performance on multiple adaptation tasks with only unlabeled target data, which verifies its effectiveness in this challenging setting.

1. Introduction

Although deep neural networks have achieved state-of-the-art performance on various visual recognition tasks [24, 17], the promising performance heavily relies on the availability of sufficient labeled dataset with diverse visual variations [7], and the training and test data should be independent and identically distributed. When the test environment is different from the source domain, the performance of most visual systems will be seriously degraded. This is known as the domain shift [42], as illustrated in Fig. 1. Domain shift is one of the key factors that prevent the transfer

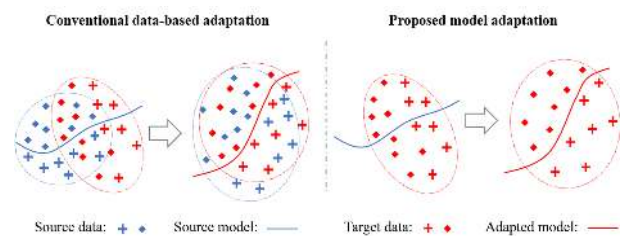


Figure 1. Comparison between conventional data-based adaptation (left) and our model adaptation (right). Conventional unsupervised domain adaptation methods require labeled source data during adaptation, while our proposed model adaptation method only relies on unlabeled target data.

of research results into real-world applications. An intuitive strategy is to re-collect and annotate sufficient target dataset to re-train or fine-tune the model [62, 35]. However, this solution is not only expensive but also not practical for manual annotation in various environments.

There are great interests in developing a visual recognition model that generalize well to different domains with few or no human annotations. Recently, unsupervised domain adaptation has received a lot of attention, since it makes large progresses in generalizing the pre-trained prediction model to the target domain where labels are not available. This is achieved by exploiting knowledge from sufficiently labeled source data [56, 12, 34]. Existing unsupervised domain adaptation methods normally assume that the source dataset is available during training. However, this assumption is not always practical in the following cases:

1. For many companies, they will only provide the learned models instead of their customer data due to data privacy and security issues.
2. The source datasets like videos or high-resolution images may be so large that it is not practical or convenient to transfer or retain them to different platforms.

Therefore, developing an unsupervised domain adapta-

tion method without source dataset has a high practical value [6]. Recent domain adaptation methods are categorized into two groups: 1) learning domain-invariant features by minimizing a specific distribution distance between the source and target domains [31]; and 2) translating the source data to the target data directly based on generative adversarial networks (GAN) [13]. Despite their great progresses, these methods cannot handle the setting where the source dataset is not available since estimating the source distribution or transformation between two domains is impossible.

In this paper, we focus on unsupervised domain adaptation without source data, referred to as unsupervised model adaptation, and follow the standard assumption where both domains share the same label space. The idea of model-based unsupervised domain adaptation is shown in the right of Fig. 1. Specifically, conventional data-based unsupervised domain adaptation aims to learn a prediction model C to generalize to the target domain based on labeled source data $\mathcal{D}_s = \{X_s, Y_s\}$ and unlabeled target data $\mathcal{D}_t = \{X_t\}$, while our model-based adaptation is to adapt the pre-trained source model C to the target domain only with \mathcal{D}_t . In other words, \mathcal{D}_s is not accessible during the model adaptation. It is noteworthy that we can easily obtain the pre-trained C with \mathcal{D}_s . However, this process cannot be reversed. Therefore, our new model-based adaptation is designed for above scenarios, which adapts an existing model to new domains.

First, to be independent of source data, we develop a Collaborative Class Conditional Generative Adversarial Networks (3C-GAN) for producing target-style training samples. To this end, a discriminator is introduced to match the target distribution by adversarial training. A class conditional generator is imposed with a semantic similarity constraint, which collaborates with the prediction model during the adaptation. Second, we introduce a weight regularization that encourages the prediction model to be close to the original source model, which can stabilize training and improve the performance. Moreover, a clustering-based regularization is incorporated into the overall objective to force the decision boundaries to be located in the low-density regions, thereby improving the final adaptation performance.

We conduct extensive experiments on multiple unsupervised domain adaptation benchmarks. In addition, ablation studies are performed to analyze contributions of each component in our model. The experimental results have verified the superiority of our method. We summarize contributions of this work as follows:

- We consider a novel and challenging adaptation setting which aims to transfer a prediction model across different domains with only unlabeled data. This is not feasible for the existing adaptation approaches.
- To avoid relying on source data, we propose 3C-GAN where the generator and the prediction model can be collaboratively enhanced during adaptation.

- We demonstrate that the proposed model is sufficiently effective on multiple domain adaptation benchmarks, and outperforms recent state-of-the-art results in the absence of source data.

2. Related Work

In this section, we focus on recent unsupervised domain adaptation methods based on Convolutional Neural Networks (CNNs) due to its superior performance.

Most domain adaptation methods mitigate the distribution discrepancy between domains according to [1]. The expected error on the target domain is bounded by: 1) the expected error on the source domain; 2) the domain discrepancy between the source and target domains; and 3) a shared expected loss which is expected to be small [60]. The expected error on the source domain can be minimized by using labeled data in the source domain. Thus the core task becomes to minimize the discrepancy between domains. Deep Domain Confusion (DDC) [58] and Deep Adaptation Networks (DAN) [31] adopt maximum mean discrepancy [15] on the final multiple layers to enforce the distribution similarity between source and target features. Joint Adaptation Networks (JAN) [34] uses the joint maximum mean discrepancy to align the joint distributions among multiple layers. Deep CORAL [54] use feature covariance to measure the domain discrepancy. Philip *et al.* [16] enforce the associations of similar features within two domains. In addition to these methods of measuring distribution discrepancy, maximizing the domain confusion via adversarial training can be used to align distributions. Domain Adversarial Neural Network (DANN) [11] introduces a domain classifier and renders the extracted features from two domains indistinguishable by a gradient reversal layer [10]. These adversarial training based methods show effective adaptation performances [3, 32]. Pinheiro *et al.* include the adversarial loss and a similarity-based classifier [45] to improve the model generalization. To integrate category information into the learning of domain-invariant features, Multi-Adversarial Domain Adaptation (MADA) [43] adopts multiple domain discriminators which correspond to each category. In [49], instead of relying on a domain discriminator, Saito *et al.* propose two task classifiers to align distributions by minimizing their discrepancy. [26] adopts sliced Wasserstein metric to measure the dissimilarity of classifiers.

Inspired by GAN [13], recent works achieve feature distribution alignment based on a generative model. Sankaranarayanan *et al.* propose a GenerateToAdapt model [50] which induces the extracted source or target embeddings to produce source-like images, such that the extracted features are expected to be domain-invariant. DuplexGAN [19] uses two discriminators for two domains to ensure that the extracted features can generate images on both domains based on a domain code. Image-to-image translation [21] pro-

vides a new direction for domain adaptation, which achieves the distribution alignment in the data space. In the absence of paired domain data, preserving the content will be non-trivial, and several recent works perform unsupervised image-to-image translation by including an extra constraint between input and the transformed output. SimGAN [51] employs a reconstruction loss between them, while PixelDA [2] and DTN [55] encourage the output to have the same class label and the semantic features as input, respectively. CoGAN [30] and UNIT [29] learn a feature space based on shared or non-shared strategies to perform cross-domain generation. Zhu *et al.* propose CycleGAN [65] which involves bi-directional translations with a cycle-consistency loss, which enforces the condition that the translated image can be mapped back to input. DiscoGAN [22] and DualGAN [61] share the same idea and achieve promising unsupervised image translation performance. CyCADA [18] is based on CycleGAN and delivers good performance on multiple domain adaptation tasks.

Additionally, some works further explore using unlabeled target data to improve generalization by co-training [59], pseudo-labeling [48, 66], and entropy regularization [52]. Some recent works focus on open set adaptation problems [63]. However, these works require source data during adaptation. Thus, most previous works are not applicable to the proposed model adaptation problem. Some incremental learning works [8, 27] are relevant to us, but they need labeled target data for new tasks. In this paper, we propose to simply use the unlabeled target dataset to adapt the pre-trained model to the target domain.

3. Proposed Method

In this section, we elaborate our model for unsupervised model adaptation problem, where we merely have access to the pre-trained prediction model C from the source domain and unlabeled target dataset X_t . Our goal is to adapt C to the target domain with X_t .

To this end, we propose a Collaborative Class Conditional Generative Adversarial Networks (3C-GAN) for model adaptation in absence of source data. Except for the existing pre-trained C , our framework consists of another two components: a discriminator D for matching target distribution and a generator G conditioned on randomly sampled labels for producing valid target-style training samples. By incorporating the generated data during training, the performance of C is improved on the target domain which can in turn promote the generation process of G . Besides, we design two regularization terms to prevent the adapted model far away from the pre-trained source model and improve the generalization on the target domain, respectively. The architecture is illustrated in Fig. 2. D , G and C are parameterized by θ_D , θ_G and θ_C , respectively. The details for each proposed component are introduced as follows.

3.1. Collaborative Class Conditional GAN

To avoid using source data for domain adaptation, we propose the Collaborative Class Conditional GAN (3C-GAN) for collaboratively improving the generator G and the prediction model C . As shown in Fig. 2, this is achieved by integrating C into the GAN framework. Different from standard GAN model, where G is only conditioned on a noise vector z , our G is further conditioned on a pre-defined label y , *i.e.*, $x_g = G(y, z)$. Also in contrast to traditional conditional GAN [37] where D is trained to distinguish real and fake pairs in a supervised manner, our D is optimized to distinguish x_t from x_g . The objective function for D can be expressed as follows:

$$\max_{\theta_D} \mathbb{E}_{x_t \sim \mathcal{D}_t} [\log D(x_t)] + \mathbb{E}_{y, z} [\log(1 - D(G(y, z)))]. \quad (1)$$

Meanwhile, G is updated to fool D by generating x_g with a similar distribution as x_t . Thus, the adversarial loss ℓ_{adv} of G can be formulated as follows:

$$\ell_{adv}(G) = \mathbb{E}_{y, z} [\log D(1 - G(y, z))]. \quad (2)$$

Although ℓ_{adv} simulates the target distribution, it cannot guarantee the semantic similarity to the input label y .

Inspired by [5], we propose a semantic similarity loss ℓ_{sem} based on the existing prediction model C . It enforces the semantic similarity between x_g and the input label y based on the prediction model C , as defined below:

$$\ell_{sem}(G) = \mathbb{E}_{y, z} [-y \log p_{\theta_C}(G(y, z))], \quad (3)$$

where $p_{\theta_C}(\cdot)$ indicates the class probability predicted by the prediction model C . ℓ_{sem} enables the generation semantics. After including ℓ_{adv} that matches the target distribution, the optimization objective of generator G is defined as follows:

$$\min_{\theta_G} \ell_{adv} + \lambda_s \ell_{sem}, \quad (4)$$

where λ_s balances two losses. We alternately update D and G for optimizing Eq. (1) and Eq. (4), respectively. As a result, G can produce new target-style instances, *i.e.*, $\{x_g, y\}$, which are used to improve performance of C on the target domain. C and G collaborate with each other during training since the enhanced C can provide more accurate guidance for G and a more reliable generation can in turn improve performance of C . Therefore, the overall framework refers to collaborative class conditional generative adversarial networks.

In addition to $\ell_{gen} = \mathbb{E}_{y, z} [-y \log p_{\theta_C}(x_g)]$, we further include two regularizations to enhance the performance of C . The final optimization objective for the prediction model C can be expressed as below:

$$\min_{\theta_C} \lambda_g \ell_{gen} + \lambda_w \ell_{wReg} + \lambda_{clu} \ell_{cluReg}, \quad (5)$$

where ℓ_{wReg} and ℓ_{cluReg} denote weight regularization and clustering-based regularization. λ_g , λ_w and λ_{clu} are used to adjust relative effects of each loss. During the adaptation process, the source dataset is not used, as shown in Fig. 2.

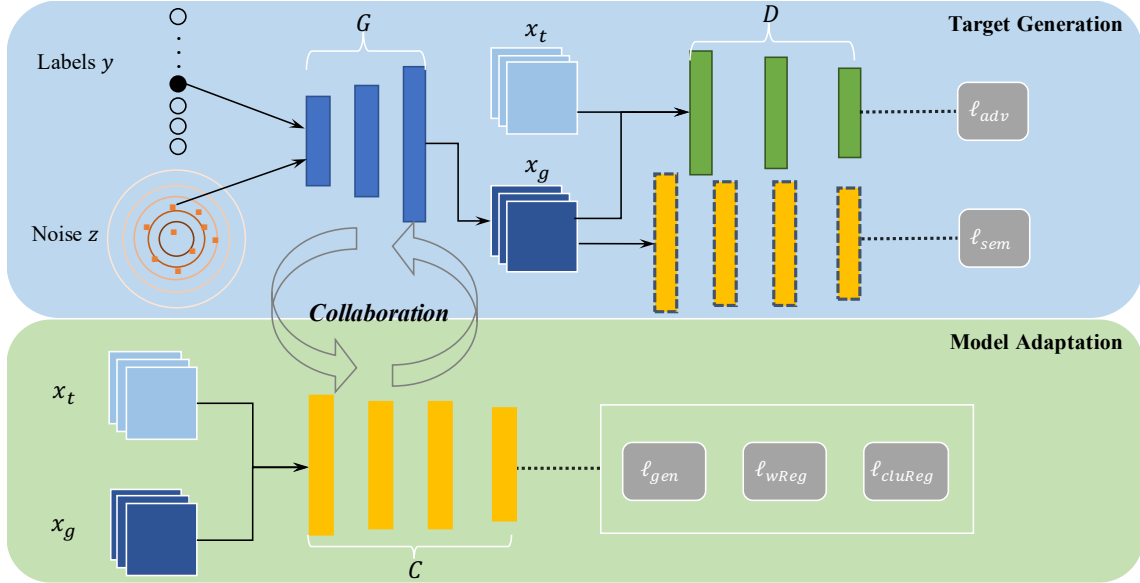


Figure 2. An overview of the proposed architecture. During target generation (top), we aim to learn a class conditional generator G for producing target-style training samples $x_g = G(y, z)$ via the discriminator D and the prediction model C (which is fixed as denoted by dashline). The generated images and proposed regularizations are used for model adaptation (bottom). These two procedures are repeated, with G and C collaborating with each other. (See text for details)

3.2. Weight Regularization

Although only incorporating the above generated target-style samples into training C can improve its performance, the training process is not always stable due to the lack of accurate supervision from the labeled source data. Inspired by [46, 57] which attempt to learn two separate but related prediction models for the source and target domains, we propose a weight regularization term ℓ_{wReg} to prevent the parameters of the prediction model C to drift far away from those of the pre-trained model learnt in the source dataset. It can be defined as follows:

$$\ell_{wReg} = \|\theta_C - \theta_{C_s}\|^2, \quad (6)$$

where θ_{C_s} is the parameters of C pre-trained on the source domain, which is fixed. We can observe that if θ_{C_s} is set to 0, ℓ_{wReg} will be reduced to standard weight decay regularization (ℓ_2). On one hand, ℓ_{wReg} prevents the adapted model from changing too significantly, which is helpful in stabilizing the adaptation. On the other hand, enforcing the adapted model similar to the source model can be regarded as preserving the source knowledge. Experiments have verified that ℓ_{wReg} leads to better adaptation in most cases.

3.3. Clustering-based Regularization

Most domain adaptation methods focus on the adaptation process, where unlabeled real target data are only used to estimate the target distribution, while we consider that unlabeled target data can be used to explore the discriminative information on the target domain. The cluster assump-

tion implies that the decision boundaries of the prediction model should not go through data regions with high density [14]. Therefore, we minimize the conditional entropy of the predicted probability on the target domain, as defined by:

$$\mathbb{E}_{x_t \sim \mathcal{D}_t} [-p_{\theta_C}(x_t) \log p_{\theta_C}(x_t)]. \quad (7)$$

However, as pointed out in [14], the conditional entropy derived in Eq. (7) is not reliable when the prediction model is not locally smooth. To improve the approximation of conditional entropy on unlabeled target data, a local smoothness constraint should be added, which is defined as follows:

$$\mathbb{E}_{x_t \sim \mathcal{D}_t} \left[\max_{\|r\| \leq \xi} \text{KL}(p_{\theta_C}(x_t) \| p_{\theta_C}(x_t + r)) \right], \quad (8)$$

where $\text{KL}(\cdot \| \cdot)$ denotes the Kullback-Leibler divergence. Following [39], we attempt to find a perturbation r that affects the prediction most within an intensity range of ξ . This constraint forces the prediction output to be similar between x_t and $x_t + r$. Consequently, the prediction model is locally smooth for each unlabeled target sample.

Therefore, the final clustering-based regularization is formulated as follows:

$$\ell_{cluReg} = \mathbb{E}_{x_t \sim \mathcal{D}_t} [-p_{\theta_C}(x_t) \log p_{\theta_C}(x_t)] + [\text{KL}(p_{\theta_C}(x_t) \| p_{\theta_C}(x_t + \tilde{r}))], \quad (9)$$

where \tilde{r} is the adversarial perturbation derived from Eq. (8).

Algorithm 1 Pseudo-code of our model adaptation process

Input: Pre-trained prediction model C on the source domain, unlabeled data X_t in the target domain, λ_g , λ_{clu} and λ_w , batch size B ;

Output: θ_C for the prediction model C ;

Initialize learning rates ζ_G , ζ_D and ζ_C for G , D and C ;

```
1: for  $epoch = 1$  to  $N$  do
2:   Randomly sample  $x_t$  of size  $B$  from  $X_t$ , and random vectors  $\{y, z\}$  from the uniform distribution;
3:   for each mini-batch do
4:     Generate new samples with  $y$  and  $z$ :  $X_g = G(y, z)$ 
5:     Update  $D$  via  $\theta_D \leftarrow \text{Adam}(\nabla_{\theta_D}(\sum_{x_t} \log D(x_t) + \sum_{y,z} \log D(1 - G(y, z))), \theta_D, \zeta_D)$ .
6:     Update  $G$  via  $\theta_G \leftarrow \text{Adam}(\nabla_{\theta_G}(\ell_{adv} + \lambda_s \ell_{sem}), \theta_G, \zeta_G)$ 
7:     if starting adaptation then
8:       Update  $C$  via  $\theta_C \leftarrow \text{Adam}(\nabla_{\theta_C}(\lambda_g \ell_{gen} + \lambda_w \ell_{wReg} + \lambda_{clu} \ell_{cluReg}), \theta_C, \zeta_C)$ 
9:     end if
10:  end for
11: end for
```

3.4. Implementation Details

Learning proceeds by alternately updating C , D and G to optimize the corresponding objectives in Eq. 5, Eq. 1 and Eq. 4, respectively. In the experiments, we do not apply ℓ_{gen} and ℓ_{cluReg} for C until the generator can produce meaningful data after several steps. The whole model is trained end-to-end and the implementation is shown in Algorithm 1.

4. Experiments

In this section, we conduct extensive experiments on multiple domain adaptation benchmarks to verify the effectiveness of our method. For each task, we only use source data to obtain the pre-trained source model, and it is not used during adaptation. The results of recent state-of-the-art domain adaptation methods are presented for comparisons or as references since most of them are not applicable when source data are not available during adaptation process.

4.1. Experimental Settings

Digit and sign datasets: we evaluate our method among five digit datasets (MNIST [25], USPS [20], MNIST-M [11], SVHN [41], Syn.Digits [11]) and two traffic sign datasets (Syn.Signs [40] and GTSRB [53]). The digit datasets contain 10 shared classes, while the traffic sign datasets contain 43 classes. Besides, Syn.Digits and Syn.Signs are synthetic domains, which is more interesting in real applications.

Office-31 [47] is a standard domain adaptation benchmark, where images are collected from three distinct domains: Amazon (**A**), Webcam (**W**) and DSLR (**D**). Three domains

share 31 classes and contain 2817, 795 and 498 samples, respectively. Following [43, 34], we evaluate on all six domain adaptation tasks. These tasks can verify the effectiveness of our method when the number of samples is small.

VisDA17 [44] is a challenging dataset for domain adaptation from synthetic domain to real domain with 12 shared classes. The synthetic domain contains around 152k images produced by rendering 3D models under different conditions. We use the validation set as the real domain, which contains around 55k images collected from MSCOCO [28]. Since the number of source data is very large, this task can demonstrate the superiority of our method which can achieve successful adaptation without source data.

For experiments on digit and sign datasets, we resize all images to $32 \times 32 \times 3$. The architecture of C is similar to the one in [52] for a fair comparison. An UpResBlock module is adopted in the generator for high-quality image generation. We adopt spectral normalization [38] in the discriminator for training stability. For experiments on Office-31 and VisDA17, we choose ResNet50 and ResNet101 [17] pre-trained on ImageNet [7] to extract features. Both generator and discriminator consist of two dense layers.

We use Adam [23] to optimize all the networks. The learning rates for D and G are 4×10^{-4} and 10^{-4} , respectively. As to C , the initial learning rates are 10^{-3} and 10^{-4} for digit/sign datasets and office-31, respectively. We decreased it 10 times during the training. For VisDA17, the learning rate is fixed with 10^{-5} . The weighting factor λ_w , λ_g and λ_{clu} are set to 10^{-4} , 10^{-1} and 1, respectively. For digit datasets, λ_{clu} is set to 10^{-1} instead.

4.2. Experimental Results

Results on digit and sign benchmarks: Table 1 compares the classification accuracy of our model adaptation and recent unsupervised domain adaptation methods. First, compared with the Source-Only model (baseline), the performance of our model on the target domain is significantly increased on all the domain adaptation tasks. In particular, the accuracy rate of our model in MNIST→MNIST-M can reach 98.5%, which outperforms the baseline by around 40%. The significant performance gains suggest that the labeled data on the source domain is not sufficient to achieve good generalization performance on the target domain, while the generated target-style training instances and regularizations in our proposed model facilitate the adaptation and largely improve the performance on the target domain. Second, all the other recent domain adaptation methods require the source data during adaptation process, while our model obtains the best or comparable performance in the absence of source data compared with the other competing methods. Specifically, the test accuracies of our model on the tasks SVHN→MNIST, USPS→MNIST and Syn.Sign→GTSRB are greater than

| Method | SVHN→MNIST | MNIST→USPS | USPS→MNIST | MNIST→MNIST-M | Syn.Digits→SVHN | Syn.Sign→GTSRB |
|-----------------|-----------------|-----------------|-----------------|---------------|-----------------|-----------------|
| Source-Only | 76.4±1.5 | 92.4±1.7 | 86.1±1.3 | 54.2±0.9 | 86.2±0.9 | 78.3±1.6 |
| DAN [31] | 71.1 | 81.1 | - | 76.9 | 88 | 91.1 |
| AssocDA [16] | 97.6 | - | - | 89.5 | 91.8 | 97.6 |
| DANN [11] | 73.8 | 85.1 | 73.0 | 77.4 | 91.1 | 88.7 |
| UNIT [29] | 90.5 | 95.9 | 93.5 | - | - | - |
| GenToAdapt [50] | 92.4±0.9 | 95.3±0.7 | 90.8±1.3 | - | - | - |
| DSN [3] | 82.7 | 91.3 | - | 83.2 | 91.2 | 93.1 |
| PixelDA [2] | - | 95.9 | - | 98.2 | - | - |
| CyCADA [18] | 90.4±0.4 | 95.6±0.2 | 96.5±0.1 | - | - | - |
| SimDA [45] | - | 96.4 | 95.6 | 90.5 | - | - |
| MCD [49] | 96.2±0.4 | 94.2±0.7 | 94.1±0.3 | - | - | 94.4±0.3 |
| VADA [52] | 97.9 | - | - | 97.7 | 94.8 | 98.8 |
| DIRT-T [52] | 99.4 | - | - | 98.9 | 96.1 | 99.5 |
| Our Model | 99.4±0.1 | 97.3±0.2 | 99.3±0.1 | 98.5±0.2 | 95.9±0.2 | 99.6±0.1 |

Table 1. Classification accuracy (%) on digit and sign dataset. ‘-’ denotes that the results are not reported.

| Method | A→W | D→W | W→D | A→D | D→A | W→A | Average |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-------------|
| ResNet50 [17] | 68.4±0.2 | 96.7±0.1 | 99.3±0.1 | 68.9±0.2 | 65.2±0.3 | 60.7±0.3 | 76.1 |
| DAN [31] | 80.5±0.4 | 97.1±0.2 | 99.6±0.1 | 78.6±0.2 | 63.6±0.3 | 62.8±0.2 | 80.4 |
| RTN [33] | 84.5±0.2 | 96.8±0.1 | 99.4±0.1 | 77.5±0.3 | 66.2±0.2 | 64.8±0.3 | 81.6 |
| DANN [11] | 82.6±0.4 | 96.9±0.2 | 99.3±0.2 | 81.5±0.4 | 68.4±0.5 | 67.5±0.5 | 82.7 |
| ADDA [57] | 86.2±0.5 | 96.2±0.3 | 98.4±0.3 | 77.8±0.3 | 69.5±0.4 | 68.9±0.5 | 82.9 |
| JAN [34] | 86.0±0.4 | 96.7±0.3 | 99.7±0.1 | 85.1±0.4 | 69.2±0.4 | 70.7±0.5 | 84.6 |
| MADA [43] | 90.0±0.2 | 97.4±0.1 | 99.6±0.1 | 87.8±0.2 | 70.3±0.3 | 66.4±0.3 | 85.2 |
| GenToAdapt [50] | 89.5±0.5 | 97.9±0.3 | 99.8±0.2 | 87.7±0.5 | 72.8±0.3 | 71.4±0.4 | 86.5 |
| Our Model | 93.7±0.2 | 98.5±0.1 | 99.8±0.2 | 92.7±0.4 | 75.3±0.5 | 77.8±0.1 | 89.6 |

Table 2. Classification accuracy (%) on office-31 based on ResNet50 [17].

99%. On MNIST→MNIST-M and Syn.Digits→SVHN, our method obtains 98.5% and 95.9%, which are competitive to DIRT-T (98.9% and 96.1%). However, DIRT-T is based on VADA which involves source data during the first adaptation stage. Interestingly, we observe that our model can achieve 99.2% and 96.7% in terms of accuracy when including source data during training, which outperforms DIRT-T.

Results on Office-31: Table 2 shows the performances of our model and the other unsupervised domain adaptation methods. All the results are obtained with ResNet50 as the backbone. The first row shows the performance by fine-tuning on the source domain as the baseline. It is clear that our model outperforms all competing methods by a large margin. Specifically, compared to GenToAdapt [50] and MADA [43] which involve complex architectures and objective functions, our model boosts performance by around 3% and 4% on average across six adaptation tasks. In addition, our model shows superior performance on difficult adaptation tasks, *i.e.*, $A \leftrightarrow D$, $A \leftrightarrow W$. It exceeds the performance of the second best method by 4.5% on average among these four tasks.

Results on VisDA17: Table 3 shows the class-level accuracy on VisDA17 based on ResNet101. Our model significantly outperforms other unsupervised domain adaptation methods. Specifically, our model achieves 81.6% class mean accuracy with the vanilla ResNet101, and this result can be further improved with a more powerful backbone.

For example, we use an enhanced ResNet101 shown in the last row of Table 3. The accuracy is increased to 83.3%, which surpasses SimDA [45] with ResNet152 by 10.4%. Besides, self-ensembling (SE) [9] relies on data augmentation and ensemble techniques, while our model outperforms SE (with minimal augmentation) by 9.1% without data augmentation. In addition, our model does not use source data during adaptation, which is more preferable in this task when the source dataset is rather large.

4.3. Visualization Analysis

To provide insights into the collaborative mechanism in our 3C-GAN, we present the generated samples conditioned on the labels from 0 to 9. As shown in Fig. 3, each column shares the same class label, and each row shares the same noise vector. Fig. 3 (top) represents the samples produced in the early stage when C is weak on the target domain, and Fig. 3 (bottom) represents the samples produced in the late stage of adaptation. We observe that our generator can learn the class-conditional data distribution in these tasks. Besides, after incorporating generated instances into training the prediction model, the performance of the prediction model is increased (see Table 1). The enhanced prediction model can also improve the target class distribution learning within the generator. An obvious illustration is shown in Fig. 3(a). The generation quality becomes much better during the late stage when the adapted prediction model is

| Method | plane | bycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck | Average |
|-------------------------------|-------|------|------|------|-------|-------|-------|--------|-------|--------|-------|-------|-------------|
| 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 |
| DAN [31] | 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 |
| MCD [49] | 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 |
| SWD [26] | 90.8 | 82.5 | 81.7 | 70.5 | 91.7 | 69.5 | 86.3 | 77.5 | 87.4 | 63.6 | 85.6 | 29.2 | 76.4 |
| SimDA [45](ResNet152) | 94.3 | 82.3 | 73.5 | 47.2 | 87.9 | 49.2 | 75.1 | 79.7 | 85.3 | 68.5 | 81.1 | 50.3 | 72.9 |
| Self-Ensembling [9] (min aug) | 92.9 | 84.9 | 71.5 | 41.2 | 88.8 | 92.4 | 67.5 | 63.5 | 84.5 | 71.8 | 83.2 | 48.1 | 74.2 |
| Our Model | 94.8 | 73.4 | 68.8 | 74.8 | 93.1 | 95.4 | 88.6 | 84.7 | 89.1 | 84.7 | 83.5 | 48.1 | 81.6 |
| Our Model † | 95.7 | 78.0 | 69.0 | 74.2 | 94.6 | 93.0 | 88.0 | 87.2 | 92.2 | 88.8 | 85.1 | 54.3 | 83.3 |

Table 3. Class-wise accuracy (%) on VisDA17 based on ResNet101 [17]. † denotes that we use an enhanced version of ResNet101 which replaces the first 7×7 convolution with three 3×3 convolutions.

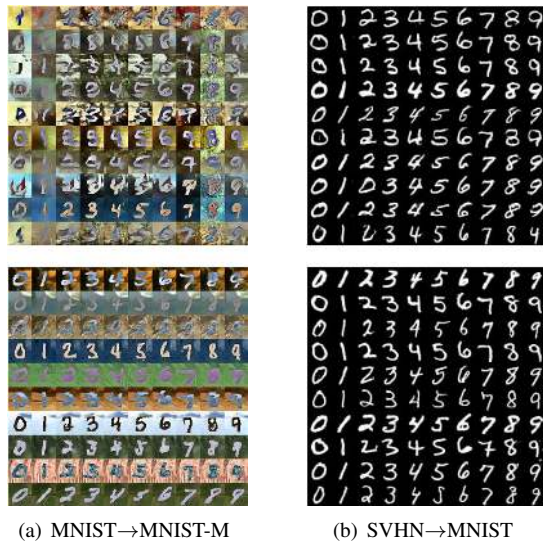


Figure 3. Class conditional generation in (a) MNIST→MNIST-M and (b) SVHN→MNIST. The top row indicates the samples generated with pre-trained source model, and the bottom row refers to the samples generated during the last adaptation stage.

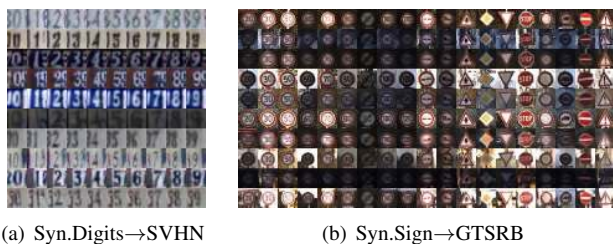


Figure 4. Class-conditional generation in (a) Syn.Digits→SVHN and (b) Syn.Sign→GTSRB (shows the first 19 out of 43 classes). Each column has the same class y and the rows share the same noise vector z .

improved on the target domain. It suggests that C and G can collaborate with each other during adaptation process.

To further demonstrate the effectiveness of our model, we visually inspect the generated images. Fig. 4 shows the class-conditional generation on two tasks. In both scenarios, the generated images are consistent with the input la-

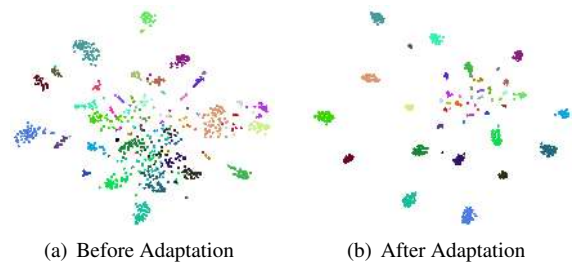


Figure 5. The t-SNE projection of the last hidden layer of target features (a) before adaptation and (b) after adaptation in the task of Syn.Sign→GTSRB. Different colors represent different classes.

| Method | SVHN | MNIST | USPS | MNIST |
|------------------------------|-----------------|-----------------|-----------------|-----------------|
| | ↓ MNIST | ↓ USPS | ↓ MNIST | ↓ MNIST-M |
| Source-Only | 68.1±1.5 | 85.3±3.1 | 71.0±1.8 | 50.3±0.7 |
| CMD [64] | 86.5 | - | 86.3 | 85.5 |
| ADDA [57] | 72.3 | 89.4 | 92.1 | 80.7 |
| CORAL [54] | 89.5 | 81.7 | 96.5 | 81.6 |
| JDDA [4] | 94.2 | - | 96.7 | 88.4 |
| <i>Our Model Variants</i> | | | | |
| w/o ℓ_{gen} | - | - | - | - |
| w/ ℓ_{gen} | 97.9±0.2 | 94.5±1.0 | 98.2±0.2 | 91.8±0.5 |
| w/ ℓ_{gen}, ℓ_{wReg} | 98.4±0.2 | 95.4±0.3 | 98.3±0.1 | 94.2±0.3 |
| Full Model | 99.2±0.1 | 97.0±0.2 | 99.3±0.1 | 97.0±0.1 |

Table 4. Ablation study on digit tasks with a small C in JDDA [4]. ‘-’ denotes the results are not reported or do not converge.

els and the style information is also encoded by the noise vector z . In addition, we visualize the distribution of target features before and after adaptation. As shown in Fig. 5, we use t-SNE [36] to project the last hidden layer features onto the 2-D space in Syn.Sign→GTSRB. The target instances are strongly clustered for each class after adaptation. These observations suggest that our model achieves accurate class-conditional generation in the target domain, which demonstrates superior model adaptation performance.

4.4. Ablation Study

To demonstrate the robustness of the proposed method, we adopt a small classifier which is similar to LeNet used in JDDA [4] for further evaluation. From Table 4, our full model still outperforms the Source-Only (baseline) by a

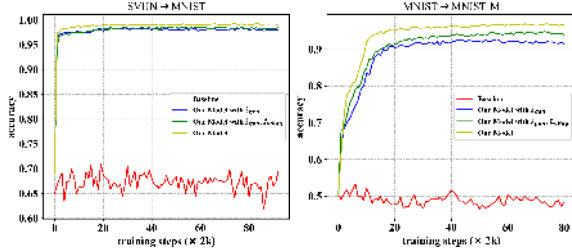


Figure 6. Comparing the performance of our model variants on the task of (a) SVHN→MNIST and (b) MNIST→MNIST-M. The accuracy is computed on the target set w.r.t. the training steps.

large margin, which achieves about or more than 30% improvement in most cases. Compared to the other unsupervised domain adaptation methods with the same classifier, our model achieves the best performance. For example, while JDDA reports an impressive performance (94.2%) on the challenging SVHN→MNIST task, our model surpasses this by about 5 percentage points. On the task MNIST→MNIST-M, our model outperforms it by about 7 percentage points. These results demonstrate the effectiveness of our model.

To explore the capability of each component, we further compare the performance of our model variants by removing the corresponding modules or loss functions.

To evaluate the contribution of the generated images in improving the model adaptation, we first remove ℓ_{gen} in our 3C-GAN. From the last block of our model variants in Table 4, the model fails to converge without ℓ_{gen} . We consider that the prediction model with only the proposed regularizations will hurt its discriminativity, due to the different distribution. Next we remove both regularizations ℓ_{wReg} and ℓ_{cluReg} . The performance of our model with only ℓ_{gen} is significantly improved from the Source-Only model, as shown in the last third row of Table 4. These results imply that our 3C-GAN can achieve reliable class-conditional generation which facilitates the model adaptation performance. Detailed illustrations of accuracy curves during training on the task of SVHN→MNIST and MNIST→MNIST-M are presented in Fig. 6. In both tasks, ℓ_{gen} is able to boost the accuracy of the baseline by a large margin, which is indicated by comparing the accuracy trends of the blue and the red curves in Fig. 6.

To investigate the effectiveness of our proposed regularization terms, we disable ℓ_{cluReg} in Eq. (5) by setting $\lambda_{clu} = 0$ during training. As shown in Table 4, the accuracy of our model by adding ℓ_{wReg} is further increased based on our model variant which only involves ℓ_{gen} . We consider that the weight regularization not only prevents the model changing significantly but also inherits the knowledge in the pre-trained source model [62]. Thus, it leads to more stable and better performance as indicated in Fig. 6 (Best viewed in color by comparing the blue and the green

| Method | A→W | A→D | D→A | W→A |
|----------------|----------|----------|----------|----------|
| w/o smoothness | 93.4±0.3 | 91.0±0.5 | 74.0±0.5 | 77.3±0.3 |
| w/ smoothness | 93.7±0.2 | 92.7±0.4 | 75.3±0.5 | 77.8±0.1 |

Table 5. Ablation study to investigate effects of the smoothness.

curves). Furthermore, by including the cluster regularization term ℓ_{cluReg} , the performance of our full model can be consistently improved by around 1 to 3 percentage points on all the tasks. In particular, as shown in the last two rows of Table 4, the accuracy increases from 94.2% to 97.0% on MNIST→MNIST-M, and 95.4% to 97.0% in MNIST→USPS. It demonstrates that our clustering-based regularization can move the decision boundaries away from the dense data regions on the target domain, which increases the generalization of the prediction model.

Furthermore, we remove the smoothness constraint of Eq. 8 to study the effect on adaptation performance. From Table 5, we observe that the accuracy dropped for the tasks A↔W and A↔D, which suggests that this constraint helps the conditional entropy estimation and improves the generalization performance.

5. Conclusion

In this paper, we propose a new model-based unsupervised domain adaptation method without source domain data. Since preparing a large amount of source data is inconvenient or even infeasible due to data privacy issues, our proposed method is more preferable for real-world applications. To this end, we propose 3C-GAN to bypass the dependence on source data. By incorporating generated images into the adaptation process, the prediction model and the generator can be mutually enhanced through collaborative learning. In addition, we introduce weight regularization and clustering-based regularization for stabilizing the training and further improving generalization performance on the target domain. We conduct extensive experiments on multiple domain adaptation benchmarks. Compared with recent data-based domain adaptation methods, our model achieves the best or comparable performance in the absence of source data, which demonstrates its effectiveness in a broad class of adaptation scenarios.

Acknowledgments. This work was supported in part by the Research Grants Council of the Hong Kong Special Administration Region (Project No. CityU 11300715), in part by the National Natural Science Foundation of China (Project No. U1611461, 61722205, 61751205, 61572199), in part by City University of Hong Kong (Project No. 7005055), in part by the Natural Science Foundation of Guangdong Province (Project No. 2016A030310422, 2016A030308013), and in part by Fundamental Research Funds for the Central Universities (Project No. 2018ZD33).

References

- [1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.
- [2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, pages 95–104, 2017.
- [3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, pages 343–351, 2016.
- [4] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI*, 2019.
- [5] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2172–2180, 2016.
- [6] Boris Chidlovskii, Stéphane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *KDD*, pages 451–460, 2016.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [8] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyang Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019.
- [9] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.
- [10] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.
- [11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17:59:1–59:35, 2016.
- [12] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, pages 597–613, 2016.
- [13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.
- [14] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2004.
- [15] Arthur Gretton, Bharath K. Sriperumbudur, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, and Kenji Fukumizu. Optimal kernel choice for large-scale two-sample tests. In *NeurIPS*, pages 1214–1222, 2012.
- [16] Philip Häusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *ICCV*, pages 2784–2792, 2017.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [18] 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. In *ICML*, pages 1994–2003, 2018.
- [19] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *CVPR*, pages 1498–1507, 2018.
- [20] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Trans. PAMI*, 16(5):550–554, 1994.
- [21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017.
- [22] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, pages 1857–1865, 2017.
- [23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012.
- [25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition, 1998.
- [26] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pages 10285–10295, 2019.
- [27] Zhizhong Li and Derek Hoiem. Learning without forgetting. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, volume 9908, pages 614–629, 2016.
- [28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.
- [29] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, pages 700–708, 2017.
- [30] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NeurIPS*, pages 469–477, 2016.
- [31] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.
- [32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1647–1657, 2018.
- [33] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, pages 136–144, 2016.
- [34] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.
- [35] Zelun Luo, Yuliang Zou, Judy Hoffman, and Fei-Fei Li. Label efficient learning of transferable representations across domains and tasks. In *NeurIPS*, pages 164–176, 2017.

- [36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [37] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.
- [38] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.
- [39] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. PAMI*, 41(8):1979–1993, 2019.
- [40] Boris Moiseev, Artem Konev, Alexander Chigorin, and Anton Konushin. Evaluation of traffic sign recognition methods trained on synthetically generated data. In *Advanced Concepts for Intelligent Vision Systems*, pages 576–583, 2013.
- [41] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshops*, volume 2011, page 5, 2011.
- [42] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct. 2010.
- [43] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, pages 3934–3941, 2018.
- [44] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshops*, pages 2021–2026, 2018.
- [45] Pedro O Pinheiro and AI Element. Unsupervised domain adaptation with similarity learning. In *CVPR*, pages 8004–8013, 2018.
- [46] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE Trans. PAMI*, 41(4):801–814, 2019.
- [47] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.
- [48] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, pages 2988–2997, 2017.
- [49] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018.
- [50] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, 2018.
- [51] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, pages 2242–2251, 2017.
- [52] Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T approach to unsupervised domain adaptation. In *ICLR*, 2018.
- [53] Johannes Stalkamp, Marc Schlipf, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks*, pages 1453–1460, 2011.
- [54] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, pages 443–450, 2016.
- [55] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017.
- [56] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015.
- [57] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.
- [58] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.
- [59] Si Wu, Jian Zhong, Wenming Cao, Rui Li, Zhiwen Yu, and Hau-San Wong. Improving domain-specific classification by collaborative learning with adaptation networks. In *AAAI*, 2019.
- [60] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5419–5428, 2018.
- [61] Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876, 2017.
- [62] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.
- [63] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *CVPR*, pages 2720–2729, 2019.
- [64] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *ICLR*, 2017.
- [65] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251, 2017.
- [66] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, pages 5981–5990, 2019.