Cycle-consistent Conditional Adversarial Transfer Networks

Jingjing Li
University of Electronic Science and
Technology of China
The University of Queensland
lijin117@yeah.net

Erpeng Chen
University of Electronic Science and
Technology of China
cep1126@163.com

Zhengming Ding
Indiana University-Purdue University
Indianapolis
zd2@iu.edu

Lei Zhu
Shandong Normal University
leizhu0608@gmail.com

Ke Lu
University of Electronic Science and
Technology of China
kel@uestc.edu.cn

Zi Huang
The University of Queensland
huang@itee.uq.edu.au

ABSTRACT

Domain adaptation investigates the problem of cross-domain knowledge transfer where the labeled source domain and unlabeled target domain have distinctive data distributions. Recently, adversarial training have been successfully applied to domain adaptation and achieved state-of-the-art performance. However, there is still a fatal weakness existing in current adversarial models which is raised from the equilibrium challenge of adversarial training. Specifically, although most of existing methods are able to confuse the domain discriminator, they cannot guarantee that the source domain and target domain are sufficiently similar. In this paper, we propose a novel approach named *cycle-consistent conditional adversarial transfer networks* (3CATN) to handle this issue. Our approach takes care of the domain alignment by leveraging adversarial training. Specifically, we condition the adversarial networks with the cross-covariance of learned features and classifier predictions to capture the multimodal structures of data distributions. However, since the classifier predictions are not certainty information, a strong condition with the predictions is risky when the predictions are not accurate. We, therefore, further propose that the truly domain-invariant features should be able to be translated from one domain to the other. To this end, we introduce two feature translation losses and one cycle-consistent loss into the conditional adversarial domain adaptation networks. Extensive experiments on both classical and large-scale datasets verify that our model is able to outperform previous state-of-the-arts with significant improvements.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision*; **Transfer learning**; **Neural networks**.

KEYWORDS

domain adaptation, adversarial training, transfer learning

1 INTRODUCTION

The amount of data we produce every day is truly mind-boggling, reported by Forbes, there are 2.5 quintillion bytes of data created each day at our current pace. These data consists of not only what we are familiar with but also novel and unseen instances. Conventional machine learning paradigms commonly assume that the test data are seen during the training stage [17, 19, 44]. Thus, it is challenging to handle the novel and unseen data in reality with conventional methods. As a practical alternative, transfer leaning [18, 21, 32] has been verified to be crucial for the success in novel environment. One of the most essential topics of transfer learning is domain adaptation [5, 26, 31, 42, 43] which investigates the problem of cross-domain knowledge transfer where the labeled source domain and unlabeled target domain have distinctive data distributions.

The very objective of domain adaptation is to reduce the domain distribution shifts between the source domain data and target domain data, so that the models trained on the well labeled source domain can be adapted to the target domain. Existing domain adaptation methods can be categorized into either shallow branch [10, 18, 21, 31, 35] or deep branch [3, 8, 13, 26]. Typically, methods in the shallow branch are built as two-step formulations. The first step is used for feature extraction and the second step focuses on domain alignment. Deep methods, however, incorporate the feature extraction and domain alignment into a united architecture. Since the feature extraction component can receive feedbacks from the domain alignment component and reinforce itself, deep approaches achieved state-of-the-art results in recent years [13, 26].

Among various deep domain adaptation methods, adversarial domain adaptation [8, 14, 26, 42] attracted a lot of attention lately. Adversarial domain adaptation, as shown in Fig. 1, introduces adversarial training into the deep architecture with the similar idea of generative adversarial networks (GANs) [11, 29]. Specifically, adversarial domain adaptation trains a domain discriminator which aims to distinguish whether a feature is from the source domain
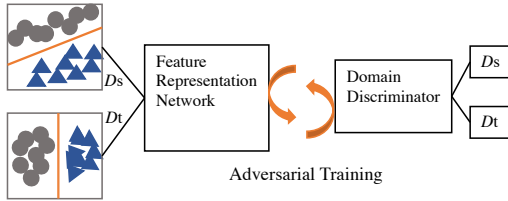
Figure 1: Adversarial domain adaptation networks. 1) The source domain D_s and the target domain D_t have different data distributions. 2) A domain discriminator is trained to distinguish source domain features from target domain features. 3) The feature representation network tries to fool the domain discriminator.

or the target domain. At the same time, a deep feature representation model tries to learn domain-invariant features to fool the domain discriminator. Once the domain discriminator fails to tell the feature sources, adversarial domain adaptation assumes that the learned features are well aligned. Then, a classifier trained on the source domain features can be directly deployed to the target domain features [37, 42].

The basic idea of adversarial domain adaptation is sound and it has also been empirically verified to be effective in many applications [3, 4, 13]. However, there is still a fatal weakness existing in current adversarial models which is raised from the equilibrium challenge of adversarial training [2]. Specifically, although most of existing methods are able to confuse the domain discriminator, they cannot guarantee that the source domain and target domain are sufficiently similar [2]. As recently claimed by Long et al. [26], adversarial adaptation methods may fail to capture the complex multimodal structures which is reflected only by the cross-covariance between the features and corresponding classifier predictions [26, 38].

In this paper, we propose a novel approach named *cycle-consistent conditional adversarial transfer networks* (3CATN) to align the two domains. At first, inspired by very recent work [26], we deploy a conditional domain discriminator by leveraging the cross-covariance of learned features and corresponding classifier predictions, so that the complex multimodal structures embedded in the data can be captured. However, since the classifier predictions are not certainty information, *deploying a strong condition from the classifier is very risky when the predictions are not sufficiently accurate.* If we put an inaccurate condition on the adversarial domain adaptation networks, the learned features may be able to confuse the domain discriminator, but there is no guarantee that the features are truly domain-invariant. Therefore, we further argue that *truly domain-invariant features should be able to be translated from one domain to the other.* This assumption does make sense by considering that *domain-invariant* actually indicates the feature space is *shared* by the two domains. In other words, the domain-invariant features are represented by the same bases and thus can be represented by each other. To this end, we train two feature translators, one translates features from the source domain to the target domain and the other translates features from the target domain to the source domain. In addition, we calculate a cycle-consistent loss by leveraging the two feature translators. With such a formulation, our method is able to not only capture the complex multimodal data structures but also avoid the negative effects caused by inaccurate conditions. In summary, we list the main contributions of this paper as follows:

1) We propose a novel deep method named *cycle-consistent conditional adversarial transfer networks* (3CATN) for domain adaptation by taking advantage of adversarial training. Compared with existing adversarial domain adaptation methods, our approach is able to not only capture the complex multimodal structures but also avoid the negative effects caused by uncertain conditions.
2) We argue that the condition with classifier predictions is risky when the the predictions are not sufficiently accurate. To address this, we further propose that truly domain-invariant features should be able to be translated from one domain to the other.
3) Experiments on both classical and large-scale datasets verify that our method is able to outperform previous state-of-the-arts with significant advantages. We show that our method is especially remarkable on relatively hard-to-transfer tasks (predictions on target data are less accurate), which verifies the claims in 2).

The rest of this paper is organized as follows. In section 2, we give a brief review of previous work reported in the community. In section 3, we present the proposed method. Section 4 reports the experiments. At last, we draw the conclusion in section 5.

2 RELATED WORK

2.1 Transfer Learning and Domain Adaptation

Domain adaptation [3, 7, 18, 22, 26, 31] is a popular branch of transfer learning [21, 32] which was proposed to handle the training data scarcity issue in supervised learning. Since most successful machine learning models [12, 16] are trained in a supervised fashion, it is challenging to handle new environment where there are no sufficient labeled data [20, 44]. At the same time, it is also unwise to build every model from scratch with plenty of related models are available [32]. Transfer learning leverages available resources to challenge new situations. In fact, the very widespread success of fine-tuning on pre-trained deep models [12, 16] has verified the effectiveness of transfer leaning.

A common assumption behind conventional machine learning algorithms is that the training set and test set have identical data distributions. However, this assumption does not always hold in real-world applications [32, 33]. Domain adaptation [34, 41] was proposed to reduce the domain shifts between the labeled source domain and unlabeled target domain. Early domain adaptation methods [1, 10, 31] focus on knowledge transfer techniques which are invariant with samples features. For instance, most of early domain adaptation approaches [6, 10, 31] learn a common subspace which is shared by the two domains, some of them [1, 15] reweight samples according to the distribution distance with target domain.

With the sweeping success of deep learning, neural networks are also deployed in domain adaptation. At first, deep domain adaptation approaches align the two domains by minimizing a pre-defined metric which measures the feature distributions, e.g., minimizing MMD distance [28], minimizing covariance distance [40]. Recently, adversarial training [8, 26, 42] was introduced into domain adaptation. The main idea of adversarial domain adaptation is to train a domain discriminator which can distinguish whether a feature is from the source domain or the target domain. The learned features would be considered as domain-invariant once the domain discriminator is confused by the learned features.
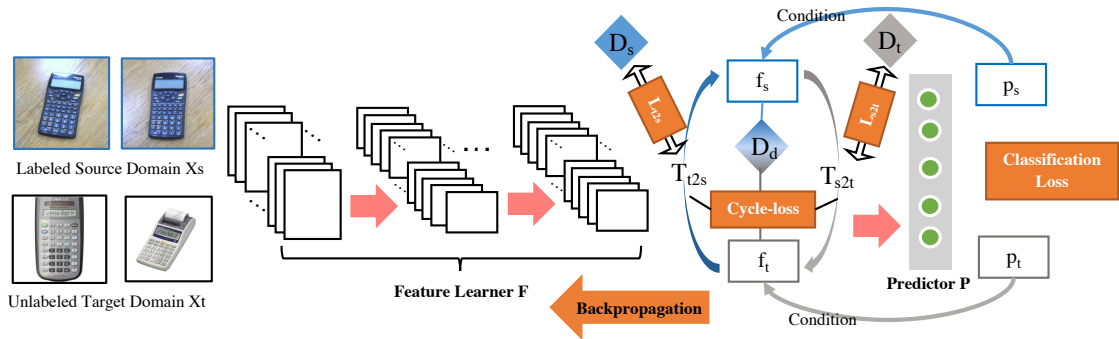
Figure 2: Idea illustration of our 3CATN. 1) A deep CNN, e.g., ResNet, is trained as a feature learner to learn domain-invariant feature representations. 2) A domain discriminator D_d is trained to distinguish source domain features from target domain features. 3) Two feature translator T_{s2t} and T_{t2s} along with their corresponding discriminators D_t and D_s are trained to translate features from one domain to the other. 4) By leveraging the two feature translators, we calculate a cycle-loss to preserve the translation consistency. 5) The classifier predictions p_s and p_t are deployed to condition the adversarial domain adaptation networks. 6) All losses are backpropagated to the feature representation network to learn truly domain-invariant features.

2.2 Adversarial Domain Adaptation

Adversarial domain adaptation [6, 10, 26, 31] is similar with generative adversarial networks (GANs) [11] but not exactly the same. GANs typically have a generator and a discriminator where the generator synthesizes fake samples from noises to fool the discriminator. Adversarial domain adaptation, however, usually does not have a "generator" which can synthesize something new. The role of "generator" in adversarial domain adaptation is played by a feature extractor. Specifically, the feature extractor learns new feature representations which is able to confuse the domain discriminator. Instead of generating from random noises in GANs, the feature extractor learns meaningful feature representations from both source domain data and target domain data.

The adversarial training can be performed at either feature level or pixel level. Feature adaptation methods [8, 26, 42] minimize the domain discrepancy on the feature space. For instance, Ganin et al. [8] optimize the standard minimax objective on the feature space. Tzeng et al. [42] optimize an inverted label objective in a unified framework. Pixel adaptation methods [3, 24] propose that the learned representations can be recovered back to raw images. As a result, pixel adaptation methods are much easier to be understood by eyes and more friendly to the end visual tasks. For instance, CoGANs [24] learns a joint distribution of multi-domain images. However, pixel adaptation methods generally cost much much more than feature-based ones. They are also sensitive to noises and missing pixels. For most classification or segmentation tasks, especially for large-scale applications, it is also unnecessary to directly handle raw pixels. Meaningful features have more discrimination power than pixels. It is worth noting that one of major differences between our method and Hoffman et al. [13] is that the latter emphasizes on pixel adaptation while our method leverages feature translation. Thus, our method is more computational efficient than [13]. In addition, as claimed by Long et al. [26], the pixel-level adaptation methods are carefully tailored to the digits and synthetic to real adaptation tasks.

In adversarial domain adaptation networks, the idea of conditional GANs [29] is widely used to promote the discriminative ability of both feature extractor and domain discriminator. For instance, Chen et al. [4] propose a global and class-specific domain adversarial learning framework for road scene segmenters. Long et al. [26] propose two novel strategies to condition the domain discriminator with classifier predictions. Our work is related with CDAN [26]. We also condition our features with the classifier predictions to preserve the multimodal data structures. However, our method is also significantly different from CDAN. The major difference is that CDAN did not consider the situation where the classifier predictions may be not sufficiently accurate (Although entropy condition is proposed in CDAN, entropy itself is an inaccurate metric to measure the classification result). As a result, putting a wrong condition on the discriminator will lead the feature extractor into risky situations. This situation is common in fine-grained visual classification tasks. For instance, a pickup truck can be mispredicted as a car. With the condition of car on the discriminator, pickups would be classified as cars for good. What is worse, the errors caused by the predictions can be propagated and magnified. In this paper, we argue that truly domain-invariant features should be able to be translated from one domain to the other. In our model, therefore, we introduce two feature translators to map features from one domain to the other. In addition, a cycle-consistent loss is calculated to guarantee that the source samples translated to target style can be translated back to the source domain. Experiments verify that such a formulation is effective for domain adaptation, especially for relatively difficult tasks where the gap between the source domain and the target domain is significantly large. In other words, compared with CDAN, our formulation is able to not only capture the complex multimodal data structures but also prevent and penalize the errors caused by classifier predictions.

3 THE PROPOSED METHOD

3.1 Notations and Definitions

In this paper, we use subscripts s and t to denote source and target, respectively. Thus, we have n_s labeled source samples $\{X_s, Y_s\}$ and

n_t unlabeled target samples X_t for unsupervised domain adaptation. The main challenge of domain adaptation is that the data distributions, both marginal distributions $\mathcal{P}_s(X_s), \mathcal{P}_t(X_t)$ and conditional distributions $\mathcal{P}_s(Y_s|X_s), \mathcal{P}_t(Y_t|X_t)$ are distinctive from the source domain to the target domain. Therefore, our goal is to train a deep network $F : x \rightarrow y$ which is able to learn discriminative domain-invariant features where the domain shifts can be sufficiently reduced. We define the related concepts as follows:

Definition 1: (Domain) A domain $\mathcal{D}$ is defined as a sample set X along with its probability distribution $\mathcal{P}(X)$, i.e, $\mathcal{D} = \{X, \mathcal{P}(X)\}$. The source domain and target domain are denoted as $\mathcal{D}_s$ and $\mathcal{D}_t$.

Definition 2: (Adversarial Discriminator) An adversarial discriminator D is defined as a binary classifier which is able to distinguish real from fake. In this paper, we introduce a domain discriminator D_d which distinguishes a source domain feature from the target domain. We also train a target sample discriminator D_t to distinguish the real target samples and fake target samples translated from the source instances, and a source sample discriminator D_s to distinguish the real source samples and fake source samples translated from the target domains.

Definition 3: (Feature Learner) A feature learner F is defined as a deep neural network which is able to learn domain-invariant feature representations for both the the source and target domains. We denote the feature representation of sample x as $f = F(x)$.

Definition 4: (Feature Predictor) A feature predictor P is defined as a classifier layer which is trained to predict possible categories of a feature representation, i.e, $p = P(f)$.

Definition 5: (Feature Translator) A feature translator T is defined as a mapping which is able to translate the features from one domain to the style of the other domain. In this paper, we train two feature translators T_{s2t} and T_{t2s}.

3.2 Overall Idea

Our cycle-consistent conditional adversarial network consists of a feature learner F, a domain discriminator D_d, a feature predictor P, two feature translators T_{s2t}, T_{t2s} along with their sample discriminators D_s and D_t.

The main goal of our model is to train an effective F which is able to learn domain-invariant feature representations, so that the classifier trained on the source domain can be applied to the target domain. To this end, we introduce the domain discriminator D_d to evaluate the feature quality of F by an adversarial manner. The feature predictor P is further added to condition the domain discriminator D_d, so that we can promote the discriminative ability of the learned features and preserve the multimodal structures embedded in the data. However, since the results of P are not always sufficiently accurate, we further argue that truly domain-invariant features should be able to be translated from one domain to the other. Therefore, we train two feature translators T_{s2t}, T_{t2s} and learn a cycle-loss. The sample discriminators D_t and D_s are used to evaluate T_{s2t} and T_{t2s}, respectively, in an adversarial manner.

For clarity, we show the core ideas of our method in Fig. 2. The base network of our 3CATN is the deep feature extractor F. Additionally, three adversarial networks are introduced to guide F. The first one is the domain discriminator D_d which conditioned by the outputs P. The other two are $\{T_{s2t}, D_t\}$ and $\{T_{t2s}, D_s\}$. Assuming

that $T_{t2s}(T_{s2t}(x_s))$ should be similar (or same in an ideal situation) with x_s, we calculate a cycle-loss to balance the loss of D_d.

3.3 The Conditional Domain Adversarial Nets

Since most of the real tasks are multi-class classification rather than binary classification, the features learned from a multi-way classification deep network naturally have multimodal structures [26]. For instance, a liger shares many characteristics of both a lion and a tiger. Thus, the visual features of a liger would have not only its specific structures but also the structures of a lion and a tiger. Such a phenomenon can be reflected by the classifier predictions, e.g., the classifier indicates the probabilities of categorizing a liger image into liger, lion and tiger are 0.8, 0.12 and 0.08, respectively. In other words, the classifier prediction p carries the possible discriminative information of the multimodal data structures. Inspired by CDAN [26], we condition both the feature learner F and the domain discriminator D_d with the classifier prediction p. As a result, we have the following minimax game:

$$
\min_{F,P} \max_{D_d} \mathcal{L}_{con} = -\mathbb{E}[\sum_{c=1}^{C} \mathbf{1}_{[y_s=c]}\log\sigma(F(x_s))]
$$
$$
+\lambda(\mathbb{E}[logD_d(\delta(\mathrm{h}_s))] + \mathbb{E}[log(1 - D_d(\delta(\mathrm{h}_t)))]),
\tag{1}
$$

where $\lambda > 0$ is a balancing parameter. In this paper, we follow CDAN and fix $\lambda = 1$ for fair comparisons. The first term in Eq. (1) is a supervised cross-entropy loss on the source domain, in which $\mathbf{1}_{[\cdot]}$ is an indicator, σ is the softmax and C is the possible categories. The second term is a conditional loss which is very similar with conditional GAN [29]. It is worth noting that $\mathrm{h} = (\mathrm{f}, \mathrm{p})$ is the joint variable of the domain specific features and corresponding classification predictions. δ is the conditioning strategy defined as follows:

$$
\delta(\mathrm{h}) = \begin{cases} \delta_{\otimes}(\mathrm{f}, \mathrm{p}) & \text{if } \dim_{\mathrm{f}} \times \dim_{\mathrm{p}} \leq 4096 \\ \delta_{\odot}(\mathrm{f}, \mathrm{p}) & \text{otherwise}, \end{cases}
\tag{2}
$$

where dim indicates the dimensionality of a vector. $\delta_{\otimes}$ is a multilinear map and $\delta_{\odot}$ is an explicit randomized multilinear map. More details of $\delta_{\otimes}$ and $\delta_{\odot}$ can be found in [26].

3.4 The Bidirectional Translation

We can learn a conditional domain adversarial network by optimizing Eq. (1) which is able to capture the multimodal data structures. However, the success of Eq. (1) builds on the assumption that the classification prediction p is sufficiently accurate. The bad news is that such an assumption does not always hold in real-world applications. If the prediction contains significant errors, the errors would be propagated to the feature learner. What is worse, the errors can be magnified with the increase of iterations.

The ultimate goal of adversarial domain adaptation is to learn domain-invariant features. However, F with flawed conditions is able to fool the domain discriminator but cannot learn truly domain-invariant features. In this paper, we argue that the truly domain-invariant features should be able to be translated from one domain to the other. To this end, we first train a feature translator T_{s2t} to translate the features from the source domain to the target domain. Specifically, we train T_{s2t} in an adversarial fashion. A discriminator D_t is simultaneously trained to distinguish real target domain

features from the translated source features. As a result, we have the follow loss function:

$$\min_{T_{s2t}} \max_{D_t} \mathcal{L}_{s2t} = \mathbb{E}[logD_t(f_t)] + \mathbb{E}[log(1 - D_t(T_{s2t}(f_s)))]. \quad (3)$$

For convenience, let $\hat{f}_t = T_{s2t}(f_s)$. Since $\hat{f}_t$ is translated from f_s, it is expected that $\hat{f}_t$ should have the same class information with f_s. Thus, we further introduce a supervised classification loss on $\hat{f}_t$ to preserve the semantic consistency and rewritten $\mathcal{L}_{s2t}$ as:

$$\min_{P,T_{s2t}} \max_{D_t} \mathcal{L}_{s2t} = \mathbb{E}[logD_t(f_t)] + \mathbb{E}[log(1 - D_t(T_{s2t}(f_s)))] \\ - \beta\mathbb{E}[\textstyle\sum_{c=1}^{C} \mathbf{1}_{[\hat{y}_t=c]}log\sigma(\hat{f}_t)]. \quad (4)$$

Similarly, we can also train a mapping from the the other direction. Specifically, we further train T_{t2s} in an adversarial fashion. A discriminator D_s is simultaneously trained to distinguish real source domain features from the translated target features. As a result, we have the follow loss function:

$$\min_{T_{t2s}} \max_{D_s} \mathcal{L}_{t2s} = \mathbb{E}[logD_s(f_s)] + \mathbb{E}[log(1 - D_s(T_{t2s}(f_t)))]. \quad (5)$$

3.5 The Cycle-consistent Loss

Inspired by the recent work on image-to-image translation [45], we encourage our model to preserve the original data information. For instance, preserving the information of f_s when translating it to $\hat{f}_t$. Let us explain this consideration by an example. The handwritten digit 1 can be similar to 7 if the upper part of 1 is distorted during the translation. However, if we preserve the original information of 1, such a error would not happen. In our model, therefore, we deploy a cycle-consistent loss to keep the consistency of original features before and after the translation. Formally, we expect that $T_{t2s}(T_{s2t}(f_s)) \approx f_s$ and $T_{s2t}(T_{t2s}(f_t)) \approx f_t$. As a result, we define the cycle-loss as:

$$\min_{T_{t2s},T_{t2s}} \mathcal{L}_{cyc} = \mathbb{E}[\|T_{t2s}(T_{s2t}(f_s)) - f_s\|_2^2] \\ + \mathbb{E}[\|T_{s2t}(T_{t2s}(f_t)) - f_t\|_2^2], \quad (6)$$

3.6 Overall Objective Function

By considering above all discussions on conditional adversarial training, bidirectional translation and cycle-consistency, we have the following loss for our 3CATN model:

$$\mathcal{L}_{3CATN} = \mathcal{L}_{con} + \eta_1(\mathcal{L}_{s2t} + \mathcal{L}_{t2s}) + \eta_2\mathcal{L}_{cyc}, \quad (7)$$

where $\eta_1, \eta_2 > 0$ are two balancing parameters to control the contribution of each part. Since we claim that truly domain-invariant features should be able to be translated from one domain to the other, we deploy the same weight for $\mathcal{L}_{z2t}$ and $\mathcal{L}_{t2s}$ in this paper to show the equality of the bidirectional translations. With the overall loss, we have our overall objective function:

$$\min_{F,P,T_{t2s},T_{t2s}} \max_{D_d,D_s,D_t} \mathcal{L}_{3CATN}. \quad (8)$$

Once the model is well-trained, we can deploy the feature learner F to learn domain invariant features for both the source and target domains. Then, the classifier trained on $F(X_s)$ can be directly used to handle $F(X_t)$.

4 EXPERIMENTS

4.1 Datasets Description

MNIST, **USPS** and Street View House Numbers (**SVHN**) [30], are three widely used handwritten digits dataset. MNIST consists of 60,000 training samples and 10,000 test samples. USPS is comprised of 7,291 training samples and 2,007 test samples. SVHN contains over 600,000 labeled digits cropped from Street View images.

Office-31 [35] is the most popular benchmark in domain adaptation. It consists of three subsets, i.e., **Amazon** (A), **Webcam** (W) and **DSLR** (D). Specifically, the images in amazon is downloaded from amazon.com. Webcam and DSLR contain images captured by a web camera and a digital SLR camera, respectively. In total, there are 4,652 samples from 31 categories in office-31 dataset.

VisDA [33] classification dataset is the currently largest dataset in the community. It consists of a training domain, a validation domain and a test domain. In total, the VisDA dataset contains over 280K images from 12 classes. In this paper, we follow the same settings in previous works [34, 36] and use the training set as the source domain and the validation domain as the target domain.

4.2 Experiment Protocols

Our model mainly consists of two parts. The first part is inspired by CDAN [26] to take advantage of conditioning with classifier predictions. However, it is easy to figure out that an inaccurate condition will lead to negative results. Thus, the second part of our model is a cycle-consistent translation to prevent the negative results. We argue that truly domain-invariant features should be able to be translated from one domain to the other. Apparently, CDAN can be regarded as a baseline of our method. Furthermore, CDAN is published very recently. The results of CDAN represent state-of-art performance in the community. As a result, we mainly compare our method with CDAN. For fair comparisons, we keep exactly the same experiment protocols with CDAN [26]. Our codes are available at github[1].

In this paper, we deploy the standard unsupervised domain adaptation settings which were widely used in previous work [10, 26]. Labeled source domain samples $\{X_s, Y_s\}$ and unlabeled target domain samples X_t are used for training. The reported results are the classification accuracy on target samples: $accuracy = |x : x \in X_t \wedge \hat{y}_t = y_t|/|x : x \in X_t|$, where $\hat{y}_t$ is the predicted label of the target domain generated by our model, and y_t is the ground truth label vector.

For digits recognition on MNIST, USPS and SVHN, we use the same setting in CDAN. Specifically, 60000, 7291 and 73257 images from MNIST, USPS and SVHN, respectively, are used for training. The basic network structure for digits recognition is similar with LeNet. We set the batch size to 224 and the learning rate as 10^{-3}.

For object recognition on Office-31, we also deploy the same setting in CDAN. The basic network for feature extraction is ResNet-50 [12] which is pre-trained on ImageNet. The batch size is 32 and the learning rate is 10^{-3}. We optimize our model by mini-batch stochastic gradient descent with a weight decay of 5×10^{-4} and momentum of 0.9. For the experiments on VisDA 2017, we also deploy ResNet-50 as the base architecture.

---

[1] github.com/lijin118/3CATN

Table 1: Results (%) of digits recognition. The best results are highlighted by bold numbers. The compared methods have the same experimental settings except for UNIT on SVHN →MNIST which uses a larger training set. The results of the baselines are cited from corresponding papers.

Method	MNIST→USPS	USPS→MNIST	SVHN →MNIST
Source only	82.2 ± 0.8	69.6 ± 3.8	67.1 ± 0.6
DANN [7]	-	-	73.6
ADDA [42]	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8
DRCN [9]	91.8 ± 0.1	73.7 ± 0.1	82.0 ± 0.1
RevGrad [8]	89.1 ± 0.2	89.9 ± 0.3	−
CoGAN [24]	91.2	89.1	−
UNIT [23]	95.9	93.6	90.5
CDAN [26]	95.6	98.0	89.2
3CATN [Ours]	$\mathbf{96.1 \pm 0.2}$	$\mathbf{98.3 \pm 0.2}$	$\mathbf{92.5 \pm 0.3}$
Target Supervised	96.3	99.2	99.2

The feature translators are implemented by one fc layer and three conv layers. The discriminator D_s and D_t are implemented by two conv layers and one fc layer. The domain discriminator D_d is implemented by three fc layers. All the hyper-parameters of this paper are tuned by importance weighted cross validation [39]. The sensitivity of parameters are discussed later in this section.

Traditional methods with deep features: TCA [31] and GFK [10]. Deep CNN methods: DANN [7], DDC [43], DAN [25], DRCN [9], JAN [28] and SimNet [34]. GAN methods: ADDA [42], JAN-A [28], RevGrad [8], CoGAN [24], MCD [36] and CDAN [26] are used for comparison. The results of some compared methods are cited from JAN and CDAN.

4.3 Digits Recognition

The results of hand-written digits recognition are reported in Table 1. We also report the results on source only as a baseline. The source only means the model is trained on only the source data and then directly applied on the target data. The other compared methods are mainly based on GANs. From the results we can see that our method is able to outperform previous state-of-the-arts. Specifically, we achieved 0.5%, 0.3% and 3.3% accuracy improvement on MNIST→USPS, USPS→MNIST and SVHN →MNIST, respectively. It is worth noting that the improvement is very hard to achieve since previous state-of-the-arts are very close to the results of target full supervised setting. Nevertheless, we achieve a significant improvement on the hardest task SVHN →MNIST.

CDAN performs very well when the classifier predictions are sufficiently accurate. The results on the hand-written digits are already sufficiently accurate for most recent works. Thus, our advantages are not quite clear on these evaluations. We perform comparably with the target supervised model. As we claimed before, our model would significantly improve CDAN when the classifier perditions are inaccurate. This is somewhat proved by the results on SVHN →MNIST. It will be further verified in following evaluations. In conclusion, our method is able to take full advantage of classifier predictions condition when they are accurate. At the same time,

we can also avoid the negative effects when the classifier conditions are inaccurate. Our approach achieves a dynamic balance by maximizing the advantages and minimizing the shortages.

It is also worth noting that the entropy conditioning in CDAN is designed to handle the issue of inaccurate classifier predictions. However, sample entropy does not necessarily indicate the classification accuracy. A strong evidence is that CDAN+E (CDAN with entropy conditioning) does not show significant advantage against vanilla CDAN. For instance, the average accuracies of CDAN+E and CDAN on Office-31 are 87.7% and 86.6%, respectively. The results of CDAN reported in this paper are all from CDAN+E since it is generally better than CDAN. We did not specific CDAN+E from CDAN for the sake of simplicity and clarity.

4.4 Objects Recognition on Office-31

The results of different methods on Office-31 are reported in Table 2. It is worth noting that we report two average results, e.g., Average 1 and Average 2. Average 1 is the average over all six evaluations. Average 2 is the average over four evaluations which excludes W→D and D→W since the two have very high accuracy for almost every method. As a result, Average 2 can reflect the performance on relatively harder tasks. From the results, we can have the following observations:

1) Transfer learning methods perform much better than the baseline ResNet-50 which is trained on only the source domain data. It proves that domain adaptation is able to minimize the domain shift and it is practical for real-world applications.

2) End-to-end deep methods outperform two-step methods which split feature extraction and knowledge adaptation. By simultaneously optimizing the two parts, the feature learner is able to receive the feedback from latter layers. The feedback can benefit the feature learner to learn more domain-invariant features.

3) Compared with previous end-to-end deep models, adversarial domain adaptation achieves state-of-the-art performance recently. Comparing JAN with CDAN, although both of them use the same base architecture, it is obvious that a conditional domain discriminator can significantly improve the domain adaptation performance.

4) Comparing average 1 and average 2, we can see that most methods have a nearly overall average accuracy since the results on W→D and D→W are very high, the two soft the overall average on tough tasks. However, the baseline ResNet also performs very well on the two evaluations. Thus, average 2 is better to reflect the performance on hard tasks of Office-31. Regarding average 2, we can see that CDAN really improves state-of-the art, which verifies that leveraging the condition with classifier prediction is quite effective. Our method is also benefited from the classifier predictions. At the same time, we can also observe that CDAN performs remarkably on A→D and A→W where the classifier predictions are relatively high (over 80% for JAN). However, for the cases D→A and W→A where the classifier predictions are relatively poor (below 70% for JAN), CDAN has no advantage against previous state-of-the-art approaches. It is even worse than GTA on both D→A and W→A. This comparison exactly verifies that the condition with classifier predictions is not reliable when the predictions are inaccurate.

5) Although we deployed the same condition with CDAN, our model is able to pervert the negative effects caused by inaccurate

Table 2: Domain adaptation results (accuracy %) on Office-31. Where average 1 is the overall average and average 2 is the average over 4 challenging evaluations except for W→D and D→W. The best results are highlighted by bold numbers.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Average 1	Average 2
ResNet [12]	68.9 ± 0.2	68.4 ± 0.2	62.5 ± 0.3	96.7 ± 0.1	60.7 ± 0.3	99.3 ± 0.1	76.1	65.1
TCA [31]	74.1 ± 0.0	72.7 ± 0.0	61.7 ± 0.0	96.7 ± 0.0	60.9 ± 0.0	99.6 ± 0.0	77.6	67.4
GFK [10]	74.5 ± 0.0	72.8 ± 0.0	63.4 ± 0.0	95.0 ± 0.0	61.0 ± 0.0	98.2 ± 0.0	77.5	67.9
DDC [43]	76.5 ± 0.3	75.6 ± 0.2	62.2 ± 0.4	96.0 ± 0.2	61.5 ± 0.5	98.2 ± 0.1	78.3	69.0
DAN [25]	78.6 ± 0.2	80.5 ± 0.4	63.6 ± 0.3	97.1 ± 0.2	62.8 ± 0.2	99.6 ± 0.1	80.4	71.4
RevGrad [8]	79.7 ± 0.4	82.0 ± 0.4	68.2 ± 0.4	96.9 ± 0.2	67.4 ± 0.5	99.1 ± 0.1	82.2	74.3
MCD [36]	74.5 ± 0.6	68.3 ± 0.2	49.9 ± 0.5	90.7 ± 0.8	43.5 ± 0.5	98.3 ± 0.5	70.9	59.1
JAN [28]	84.7 ± 0.3	85.4 ± 0.3	68.6 ± 0.3	97.4 ± 0.2	70.0 ± 0.4	99.8 ± 0.2	84.3	77.2
JAN-A [28]	85.1 ± 0.4	86.0 ± 0.4	69.2 ± 0.4	96.7 ± 0.3	70.7 ± 0.5	99.7 ± 0.1	84.6	77.8
GTA [37]	87.7 ± 0.5	89.5 ± 0.5	72.8 ± 0.3	97.9 ± 0.3	71.4 ± 0.4	99.8 ± 0.4	86.5	80.3
CDAN [26]	92.9 ± 0.2	94.1 ± 0.1	71.0 ± 0.3	98.6 ± 0.1	69.3 ± 0.3	100 ± 0.0	87.7	81.8
3CATN [Ours]	**94.1 ± 0.3**	**95.3 ± 0.2**	**73.1 ± 0.2**	**99.3 ± 0.5**	**71.5 ± 0.6**	**100 ± 0.0**	**88.9**	**83.5**

classifier conditions. We argue that truly domain-invariant features should be able to translate from one domain the other. As a result, we introduce two feature translation losses and one cycle-consistent loss. The results verify that our model can perform well with good classifier conditions. At the same time, our model is also robust with bad classifier conditions. This verifies our claim that our approach is able to not only capture the complex multimodal structures but also avoid the negative effects caused by uncertain conditions.

6) In terms of numbers, we achieved 1.2% accuracy improvement w.r.t. average 1 and 1.7% accuracy improvement w.r.t. average 2. It is worth noting that we outperform state-of-the-art CDAN 2.1% and 2.2% on the two hardest tasks D→A and W→A, respectively.

4.5 Large-scale Test

We further evaluate our model on large-scale domain adaptation datasets VisDA 2017. It is worth noting that VisDA is more challenging than Office-31. Compared with Office-31, VisDA has much more images. It is also worth noting that VisDA challenge has classification and segmentation tasks. Limited by space, in this paper we only focus on the classification task. The classification results on VisDA are reported in Table 3.

The results on VisDA also verify that our method can outperform previous state-of-the-art with significant advantages. We achieve 3.2% accuracy improvement over the 12 classes in VisDA. From the results on VisDA, we can see that our method is able to perform well on large-scale datasets. It is able to maximize the advantages of accurate predictions and, at the same time, prevent the negative effects caused by inaccurate predictions.

4.6 Qualitative Evaluation

For a better understanding, we conduct a qualitative evaluation on W→A. We show some images which are misclassified by CDAN but well-handled by our method in Fig. 4. On the evaluation of W→A, there are 163 out of 2849 samples are misclassified by CDAN but well-handled by our 3CATN. At the same time, almost all the correctly classified samples by CDAN are also correctly classified by our 3CATN. There are only 15 exceptions. For the sake of fairness, we also show some of them in Fig. 4.

4.7 Model Analysis

Parameters. The parameters in our model are tuned by importance weighted cross validation [39]. In most evaluations, we fix $\lambda = 1$ as in CDAN. Since β plays the similar role with λ, we also set $\beta = 1$. The weight for feature translators η_1 is set as 0.01 and the weight for cycle-loss η_2 is set as 0.1. To fully evaluate our model, we report the parameter sensitivities of β, η_1 and η_2 in Fig. 3(a), Fig. 3(b) and Fig. 3(c), respectively. It can be seen that β is not sensitive. Our model performs stably with different values of β. η_1 and η_2 are suggested to be set smaller than 1.

Stability. Adversarial training is famous for hard of training. Therefore, we report the training stability of our model in Fig. 3(d). It can be seen that 3CATN performs stably with the increase of iterations and it usually can achieve a stable result within 10,000 iterations.

Ablation. To show the effect of each component in our model, we report the results of ablation study in Table 4. The basic network of our model is the ResNet which learns new feature representations. The result of ResNet is reported as S0. Then, we leverage the conditional adversarial training to capture the multimodal data embedded in the data. The result is reported as S1. Furthermore, two feature translators are introduced to learn truly domain-invariant features and balance the inaccurate conditioning issue, which is reported as S2. At last, a cycle-consistent loss is calculated to preserve the consistency before and after feature translation. The result is reported as S3. The results in Table 4 verify that each part in our model plays an important role for the final results.

It is worth noting that S1 is based on CDAN, S2 and S3 are our contributions. Combining the results in Table 4 and Fig. 4, we can find that CDAN is risky when the classifier predictions are not sufficiently accurate, e.g., the classifier may be confused with a phone and a calculator and putting a condition of calculator on the category phone leads to the misclassification of phone images. Our method, however, is able to alleviate the negative effects caused by inaccurate conditions. In the introduction, we argued that truly domain-invariant features should be able to be translated from one domain to the other. To this end, we introduce two feature translators and one cycle-consistent loss. The results of ablation study and qualitative evaluation verify that both translators and

Table 3: Domain adaptation results (accuracy %) on VisDA-2017 dataset. All of the base networks are ResNet-50.

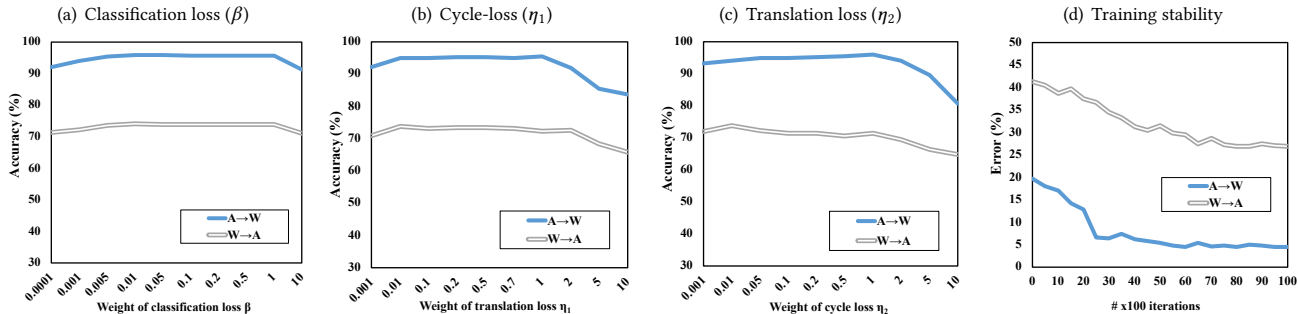| Method	ResNet[12]	DAN [25]	RTN [27]	RevGrad [8]	JAN [28]	SimNet [34]	CDAN [26]	3CATN [Ours]
Result	49.5	53.0	53.6	55.0	61.6	69.6	70.0	**73.2**

Figure 3: Parameter sensitivity (a-c) and training stability (d). The parameters in our model are tuned by importance weighted cross validation [39]. W→A and A→W on Office-31 dataset are evaluated as examples.

Figure 4: Qualitative results on W→A. Blue label is the ground-truth and gray label is the prediction. This figure mainly shows some randomly selected samples which are wrongly predicted by CDAN but correctly classified by our 3CATN. Although almost all of the samples which are correctly predicted by CDAN are also correctly handled by our method, we show several exceptions within the dash lines for the sake of fairness.

Table 4: The results of ablation study on D→A and W→A.

Settings	D→A	W→A
S0: ResNet	62.5	60.7
S1: S0+ conditional transfer loss	71.0	69.3
S2: S1 + feature translation loss	71.9	70.4
S3: S2 + cycle-consistent loss	**73.1**	**71.5**
S4: S3 - conditional transfer loss	67.5	66.9

cycle-loss definitely benefit the model. For instance, we can see that feature translators and cycle-loss boost the overall performance on D→A with 0.9% and 1.2%, respectively.

5 CONCLUSION

In this paper, we proposed a novel adversarial domain adaptation method 3CATN. 3CATN takes advantage of conditional adversarial training. In order to address the inaccurate conditioning issue, we argue that truly domain-invariant features should be able to be translated from one domain to the other. As a result, two feature translators and one corresponding cycle-consistent loss are introduced into conditional adversarial domain adaptation networks. Extensive experiments verify that our method is able to outperform previous state-of-the-art methods with significant advantages. In our model, we address the inaccurate conditioning issue with external forces, e.g., balancing the weight between classifier condition and feature translation. In our further work, we will study handling the problem from internal side, i.e., explicitly evaluating the accuracy of classifier predictions.

6 ACKNOWLEDGMENTS

REFERENCES

[1] Rahaf Aljundi, Rémi Emonet, Damien Muselet, and Marc Sebban. 2015. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *CVPR*. 56–63.

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*. JMLR. org, 224–232.

[3] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, Vol. 1. 7–16.

[4] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. 2017. No more discrimination: Cross city adaptation of road scene segmenters. In *CVPR*. 1992–2001.

[5] Zhengming Ding, Nasser M Nasrabadi, and Yun Fu. 2018. Semi-supervised Deep Domain Adaptation via Coupled Neural Networks. *IEEE TIP* 27, 11 (2018), 5214–5224.

[6] Zhengming Ding, Ming Shao, and Yun Fu. 2014. Latent Low-Rank Transfer Subspace Learning for Missing Modality Recognition.. In *AAAI*. 1192–1198.

[7] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014).

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

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*. Springer, 597–613.

[10] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*. IEEE, 2066–2073.

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

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *ICML*. 1994–2003.

[14] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. 2018. Conditional generative adversarial network for structured domain adaptation. In *CVPR*. 1335–1344.

[15] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. 2016. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*. 5081–5090.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

[17] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. 2019. Leveraging the Invariant Side of Generative Zero-Shot Learning. In *CVPR*. 7402–7411.

[18] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, and Heng Tao Shen. 2019. Locality Preserving Joint Transfer for Domain Adaptation. *IEEE TIP* (2019).

[19] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From Zero-Shot Learning to Cold-Start Recommendation. In *AAAI*.

[20] Jingjing Li, Ke Lu, Zi Huang, and Heng Tao Shen. 2017. Two birds one stone: on both cold-start and long-tail recommendation. In *ACM MM*. ACM, 898–906.

[21] Jingjing Li, Ke Lu, Zi Huang, Lei Zhu, and Heng Tao Shen. 2018. Transfer Independently Together: A Generalized Framework for Domain Adaptation. *IEEE TCYB* (2018).

[22] Jingjing Li, Jidong Zhao, and Ke Lu. 2016. Joint Feature Selection and Structure Preservation for Domain Adaptation.. In *IJCAI*. 1697–1703.

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

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

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. In *ICML*. 97–105.

[26] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *NIPS*. 1647–1657.

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. In *NIPS*. 136–144.

[28] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*. 2208–2217.

[29] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, Vol. 2011. 5.

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

[32] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE TKDE* 22, 10 (2010), 1345–1359.

[33] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 2017. VisDA: The Visual Domain Adaptation Challenge. arXiv:arXiv:1710.06924

[34] Pedro O Pinheiro and AI Element. 2018. Unsupervised domain adaptation with similarity learning. In *CVPR*. 8004–8013.

[35] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *European conference on computer vision*. Springer, 213–226.

[36] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. *CVPR* (2018).

[37] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. 2018. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR 2018*. 8503–8512.

[38] Le Song, Jonathan Huang, Alex Smola, and Kenji Fukumizu. 2009. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *ICML*. ACM, 961–968.

[39] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. 2007. Covariate shift adaptation by importance weighted cross validation. *JMLR* 8, May (2007), 985–1005.

[40] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*. Springer, 443–450.

[41] Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200* (2016).

[42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *CVPR*, Vol. 1. 4–13.

[43] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[44] Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-Shot Learning-The Good, the Bad and the Ugly. *arXiv preprint arXiv:1703.04394* (2017).

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*. 2223–2232.