Domain Impression: A Source Data Free Domain Adaptation Method

Vinod K Kurmi
IIT Kanpur
vinodkk@iitk.ac.in

Venkatesh K Subramanian
IIT Kanpur
venkats@iitk.ac.in

Vinay P Namboodiri
University of Bath
vpn22@bath.ac.uk

Abstract

Unsupervised Domain adaptation methods solve the adaptation problem for an unlabeled target set, assuming that the source dataset is available with all labels. However, the availability of actual source samples is not always possible in practical cases. It could be due to memory constraints, privacy concerns, and challenges in sharing data. This practical scenario creates a bottleneck in the domain adaptation problem. This paper addresses this challenging scenario by proposing a domain adaptation technique that does not need any source data. Instead of the source data, we are only provided with a classifier that is trained on the source data. Our proposed approach is based on a generative framework, where the trained classifier is used for generating samples from the source classes. We learn the joint distribution of data by using the energy-based modeling of the trained classifier. At the same time, a new classifier is also adapted for the target domain. We perform various ablation analysis under different experimental setups and demonstrate that the proposed approach achieves better results than the baseline models in this extremely novel scenario.

1. Introduction

Deep learning models have been widely accepted in most of the computer vision tasks. These models, however, suffer from the problem of generalization due to dataset biases. As a result, a model trained on one dataset often performs poorly on other datasets [60]. Domain adaptation methods try to resolve these issues by minimizing the discrepancy between the two domains. One possible way to minimize the discrepancy is by obtaining domain invariant features. These features are such that the classifier trained on one domain performs equally well on the other domains. Domain invariant features are obtained by introducing some auxiliary tasks to minimize the distribution discrepancy of domains. To train the auxiliary task, all existing domain adaptation approaches require access to the source datasets. The source and target datasets should both be available during the adaptation process. Nevertheless, this is not always possible in several practical scenarios. The reasons could be memory storage requirements, challenges in sharing data, privacy concerns, and other dataset handling issues. For example, the popular dataset, like Image-Net, consists of nearly 14 million images requiring hundreds of gigabytes for storage. Another concern is related to the privacy of the dataset. In some cases, the sensitive dataset can not be shared to adapt the model for a new dataset. These limitations of the traditional domain adaptation models create a bottleneck to use it for the practical scenarios. Thus, assuming the availability of the source dataset is a severe issue in existing domain adaptation models.

In this paper, we propose a domain adaptation model that does not require access to source datasets at all points of time. Specifically, we assume that we have access to a classifier that is trained on the source dataset. Only the accessibility of the classifiers instead of the whole dataset makes the model utility in the practical scenarios. We utilize the pre-trained classifier via modeling it as an energy-based function to learn the joint distribution [15]. We also use a generative adversarial network (GANs) to learn the underlying data distribution of the source dataset in conjunction with this pre-trained classifier. Once the generative model is trained using the pre-trained classifier, we proceed to generate labeled data-points that can apply in the adaptation task. We thus eliminate the need for access to the source dataset during adaptation. These generated samples can be treated as a proxy samples to train the domain adaptation model. We learn a generative function from a discriminative function by modeling it as an energy-based function. The energy of it is defined with LogSumExp() values [15]. Another discriminative property of the classifier can be used with cross-entropy loss to train the generative function. Thus, the proposed method fully utilizes the information of the pre-trained classifier for the adaptation.

Figure 1 visualized the proposed domain adaptation framework. In Figure 1(a) shows the distributional mismatch between source and target domain while in (b) the dummy source samples are generated using the pre-trained classifier, and the last adaptation stage is shown in (c),

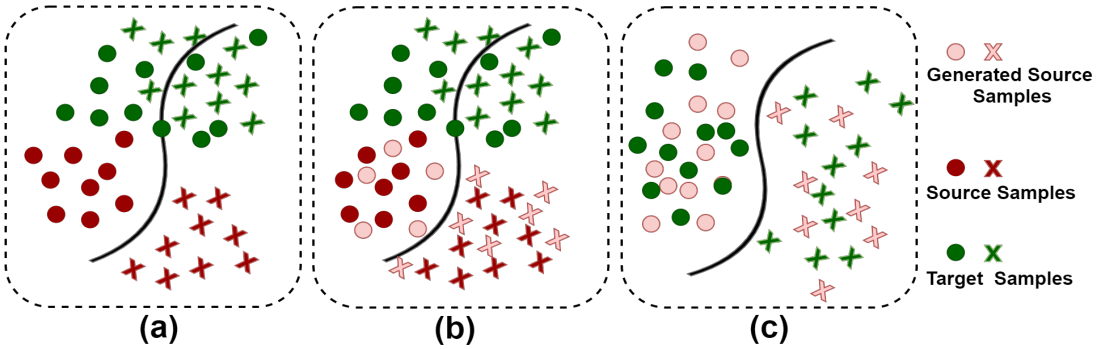

Figure 1: Illustration of proposed domain adaptation methods: (a) Without adaptation, the classifier trained on the source data can not correctly classify the target samples. (b) Proxy samples are generated using the trained classifier. (c) Adaptation of classifier using the proxy samples(generated). In the adaptation algorithms, only the proxy samples and target samples are used, source samples are never used in the adaptation process (best viewed in color).

where the classifier is adapted for the target domain using the dummy labeled samples.

The main contributions of the proposed framework are as follows:

- We provide a generative framework to tackle the source data free domain adaptation problem.
- The trained classifier is treated as an energy-based model to learn the data distribution along with a generative adversarial network.
- We show that the generated domain impression obtained using the pre-trained classifier can be applied to other existing domain adaptation methods.
- We provide detailed ablation analysis for the proposed model to demonstrating its efficacy. We also provide comparisons with the existing baselines that use full source sample information. Our method is comparable to these baselines *without* using the source samples.

2. Literature Survey

Domain adaptation has been widely studied in the literature. All the domain adaptation frameworks try to minimize the discrepancy of source and target domain [68, 52, 12, 36]. Reconstruction has been explored as in DRCN [13])and it's variants are designed to deal with two tasks, viz., classification and generation simultaneously [17, 53].

Adversarial Domain Adaptation: Adversarial methods for generating images (GANs) [14] were proposed a few months earlier to adversarial methods for domain adaptation using a gradient reversal layer (GRL) by Ganin and Lempitsky [12]. Adversarial domain adaptation was extended by other frameworks such as ADDA [61], TADA [64] and CADA [25]. These methods also suffer from the mode collapse problems. To address the mode collapse problem, multi-discriminator (MADA) [47], CD3A [24] and other types of discriminator based methods have been proposed [26, 47, 64]. Recently there are other adversarial loss based domain adaptation methods [8, 59, 33] that have been proposed to solve the domain adaptation problem more efficiently. In the drop-to-adapt method [29] leverages adversarial dropout to learn strongly discriminative features by enforcing the cluster assumption. The augmented feature-based method [63] proposes to minimize the discrepancy between two domains. A conditional GAN based model has been explored in [18] for better semantic information. A collaborative and adversarial network (CAN) [76] has been proposed through domain-collaborative and domain adversarial training of neural networks to learn domain informative features. Feature adaptation alone is not sufficient for adaptation sometimes. So classifier adaptation based methods are also introduced. Transferable adversarial training (TAT) [35] generates transferable examples to fill in the gap between the source and target domains and adversarially trains the deep classifiers. In [65], Bayesian uncertainty between source and target classifier is matched to adapt the classifier.

Privacy Concerned Domain Adaptation: There have been works presented to preserve the privacy of data in the learning process [1]. Work presented in [30, 20] deals with the privacy concerns of data in domain adaptation. These models transform the data into privacy-preserving domains using some metric like optimum transport [11]. The Federated Transfer Learning [69, 48] promises to combine multiple source data in the private mode. All the works so far, however, require access to the source data for adaptation. Source data free adaptation method for off-the-self classifier [44] improves the performance of the off-the-shelf tool in the target domain by accessing some of the labeled data for the target domain. Other source data free adaptation

methods [9] are also applicable where source data is absent, but again they assume access to some of the target labels. By utilizing the classifier's information, the model can also generate samples [15].

Adversarial Attacks: The adversarial learning framework is also well explored in the adversarial attacks and perturbations [23, 40]. These methods have been further extended for obtaining the class and data impressions [41, 42]. The knowledge of the classifier is also used for new unseen class samples [2]. A recent work [22] suggests a domain adaptation model where source data and target data never occurred together and where class boundaries are learned in the procurement stage, while adaptation occurs in the deployment stage. However, though some works aim to reduce the need for source data, no work considers the case where source data samples are not used for training, and target labels are also not available.

Generative Models: The generative approaches have successfully applied in many zero-shot recognition algorithms [66, 74]. In [21], authors generate novel examples from seen-unseen classes using the variational encoder-decoder. Other VAE based generative frameworks have been used in [58, 39]. Similarly, in [55], adversarial learning has been applied in generalized zero-shot learning. Generative adversarial network [14, 49] are very popular due to its capabilities of generating natural images and learn the data distribution efficiently. Conditional GAN [38] also applied in many application such as cross-modal [75, 67], image in painting [46, 72, 71] and colorization [43]. Very recently, work for generative data from the trained classifier is proposed in DeepInversion [70], where the statistics of the batch normalization layer are used to obtain the training data, which could enforce the constraint on the trained classifier. Similarly, the work proposed in [54] generated the images from a robust classifier. The robust classifiers are trained using the robust optimization objective [37]. Other works related to data-free distillation are resented in [7], where a student network is trained without using the data. Similarly authors of [42] propose the distillation in zero shot learning framework.

Recently, there are source data free adaptation has been presented. In [32], a generative model is used to h generated target-style data using clustering-based regularization loss. SHOT [34] uses information maximization and self supervised pseudo-labeling to implicitly align representations of target and source without accesing the source data.

3. Background: Generative model from Discriminative model

The objective of the discriminative model it to obtain the class conditional distribution $p(y|x)$, it focuses on the classification boundaries. Here x is given input and y is label. The generative models learn the joint distributions of $p(x, y)$ from the data generation process. we rewrite the log likelihood of joint distribution distribution using the Bayes theorem as [15]

$$\log p_\theta(x, y) = \log p_\theta(x) + \log p_\theta(y|x) \tag{1}$$

Here θ is the parameter of the model. The class-conditional distribution $p_\theta(y|x)$ are obtained by cross-entropy loss from the trained-classifier. The $\log\ p_\theta(x)$ can be expressed in form of energy based models [28]. We define the $\log\ p_\theta(x)$ as energy based functions as discussed in [15]. The derivative of the log-likelihood with respect to θ can be expressed as [15]

$$\frac{\partial \log p_\theta(\mathbf{x})}{\partial \theta} = \mathbb{E}_{p_\theta(\mathbf{x}')}\left[\frac{\partial E_\theta\left(\mathbf{x}'\right)}{\partial \theta}\right] - \frac{\partial E_\theta(\mathbf{x})}{\partial \theta} \tag{2}$$

Energy functions map an input x to a scalar. We define the energy function by LogSumExp$(\cdot)$ of the logits of the trained classifier similar to [15]

$$E_\theta(\mathbf{x}) = -\log \sum_{x \in P_\theta(x), y} \exp\left(P_c(\mathbf{x})[y]\right) \tag{3}$$

$P_c(\mathbf{x})[y]$ indicates y^{th} index of output of classifier $P_c(\mathbf{x})$.

4. Source Data Free Adaptation

In this section, we discuss the source data free adaptation technique using a trained classifier. This problem is divided into two parts: the first part is to obtain the samples from the classifier, we call it the *Generation module*. The second part is to adapt the classifier for the target domain, called *Adaptation module*. These two modules are shown in Figure 2.

For the generation module, we work with the conditional GAN framework [38] as a generative function to obtain the samples. The cross-entropy loss is used to obtain the domain impression and samples with class boundaries from the classifier. Note that by only the cross-entropy loss with GAN, we can enforce that generated samples follow only the conditional distribution $p(y|x)$. To learn the proxy samples of source data distribution, we model the joint distribution $p(x, y)$ defined in Eq. 1. For the adaptation module, we use the adversarial learning framework to make the feature invariant to the target domain with the generated data using a discriminator.

4.1. Problem Formulation

The source data free domain adaptation problem can be formulated as follows. We consider a classifier P_c, which is trained on the source dataset $\mathcal{D}_s$ for the classification task. The assumption or constraint is that the source dataset is

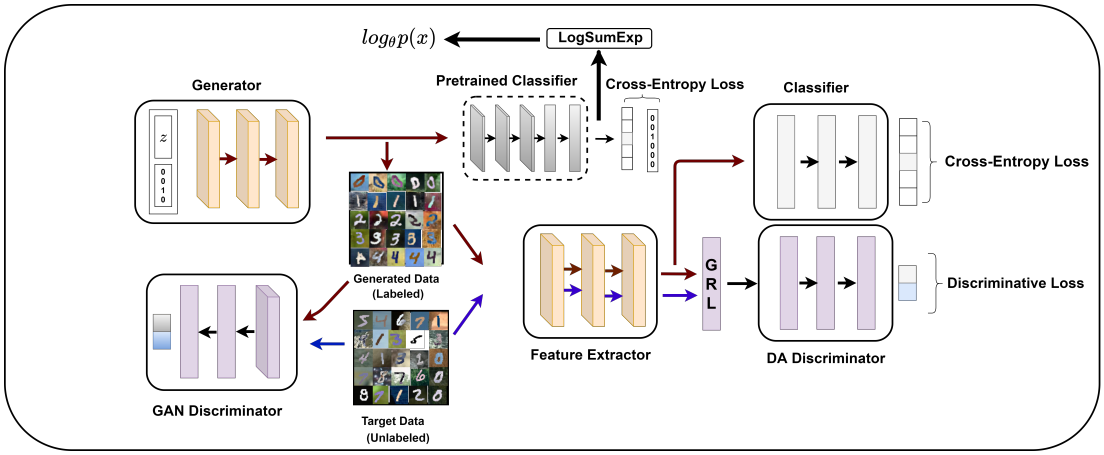

Figure 2: Source Data free Domain Adaptation: Generator (G), GAN discriminator(D_g), Feature extractor (F), Classifier (C) and Domain discriminator (D_d) are trainable while the pre-trained Classifier (P_c) is set to frozen. z is the latent noise vector. GRL is gradient reversal layer [12].

not available for adaptation. We are only provided the unlabeled target dataset $\mathcal{D}_t$ at training time. We further assume that the $\mathcal{D}_s$ comes from a source distribution $\mathcal{S}$ and $\mathcal{D}_t$ comes from a target distribution $\mathcal{T}$. We assume that there are N_t unlabeled target data points.

4.2. Proposed Model

In the proposed method, we divide the model into two parts; one is a *Generation* module, and the second one is an *Adaptation* module.

Data Generation Module: The proxy samples are obtained using a GAN framework with utilizing the source classifier. The objective is to learn the joint distribution $p(x, y)$ of the source data. The basic idea behind this approach is to obtain the samples that can be perfectly classified by the classifier. We use a parametric data generative neural network that is trained to maximize the log-likelihood defined in Eq. 1. In this equation, the first term can be maximized using the derivative defined in Eq. 2. The second term is optimized using the cross-entropy loss. A generative adversarial network, in conjunction with a trained classifier, is also applied to generate better samples. The vanilla GAN [49, 14] is an unconditional GAN and thus is not suitable here; because it is not guaranteed in the Vanilla GAN that only produce the specific desired class examples. So in the proposed generation framework, we use a conditional generative adversarial network [38], where the condition can be given as one-hot encoding and the latent noise vector to the generator to produce diverse samples. For obtaining the class-specific samples, we train this conditional generator with the cross-entropy loss of the classifier. In this case, we do not update the parameters of the pre-trained classifier; we only update the generator to produce the samples that can be classified as a given class vector. This formulation produces samples that may not be considered as natural samples, and it also produces adversarial noise examples. Thus these samples can not be used for further adaptation tasks. To obtain natural samples, we use an adversarial discriminator; it is trained with the help of target domain samples. The generator's parameters are updated with the adversarial loss from the discriminator and cross-entropy loss of the classifier.

Domain Adaptation Module: The domain adaptation module consists of a shared feature extractor for source and target domain datasets, a classifier network, and a discriminator network similar to [12]. The discriminator's objective is to guide the feature extractor to produce domain invariant features using a gradient reversal layer. In the proposed framework, the domain discriminator is trained to discriminate between the generated labeled samples and the unlabeled target samples. Similarly, we fine-tune the trained classifier for the labeled generated samples. In this module, all networks, i.e., feature extractor, classifier, and discriminator, have learnable parameters. We also have experimented with the generation and adaptation processes separately. In this variant, we first train the generative model using the likelihood and GAN objective functions. Then generative models parameters are set to be frozen and obtain samples. After that, these samples are used for adaptation. Here we have to fix the number of samples required for adaptation. The adaptation performance depends upon the number of the samples, as shown in the ablation study section in Table 4.

4.3. Loss functions

The proposed Source Data free Domain Adaptation (SDDA) model is trained with these following losses.

Likelihood based loss ($\mathcal{L}_{lik}$): The objective is to learn a joint distribution of the source data from a discriminative

model. This process required a maximize the log-likelihood of data obtained from the generative models as defined in Eq. 1. Thus loss function is written as

$$\mathcal{L}_{lik} = -\log p_\theta(\mathbf{x}) \tag{4}$$

The derivative of it is obtained from Eq. 2.

Adversarial Loss ($\mathcal{L}_{adv}$): This loss is used to train the GAN discriminator to discriminate between real data and data generated through the generator. The generator and GAN discriminator are adversaries. Here a_i is a target data, sampled from $\mathcal{T}$, y is the generated class label and z is the latent noise vector, sampled from the normal distributions P_z. Loss for the generator is defined as:

$$\mathcal{L}^g_{adv} = \sum_i \log(1 - D_g(G(\boldsymbol{z_i}, \boldsymbol{y_i}))) \tag{5}$$

Similarly loss for the GAN discriminator is defined as:

$$\mathcal{L}^d_{adv} = (\sum_i \log D_g(G(\boldsymbol{z_i}, \boldsymbol{y_i})) + \sum_{\boldsymbol{a}_i \sim \mathcal{T}} \log(1 - D_g(\boldsymbol{a_i}))) \tag{6}$$

Cross-Entropy Loss ($\mathcal{L}_{crs}$): This loss is obtained by passing the generated images to the pre-trained classifier. The predicted output of the pre-trained classifier is compared with the class vector that is input to the generator. This loss does not update the parameters of pre-trained classifier. It only updates the parameters of the generator to produce class consistent images.

$$\mathcal{C}_{crs} = \frac{1}{N_g} \sum_{g_i \in \mathcal{D}_g} \mathcal{L}_c(P_c(g_i)), y_i) \tag{7}$$

Where $g_i = G(z_i, y_i)$ is a generated image sample. $\mathcal{L}_c$ is the tradition cross entropy loss. N_g are the generated samples. P_c is the pre-trained classifier.

Domain Discriminative Loss ($\mathcal{L}_{dis}$): This loss is used to obtain domain invariant features from the feature extractor. It is a binary classification loss between the source and target samples. The discriminator is trained with the gradient of loss. In contrast, the feature extractor is trained by the negative gradient of this loss (using gradient reversal layer [12]) to obtain domain invariant.

$$\mathcal{L}_{dis} = \frac{1}{N} \sum_{x_i \in \mathcal{D}_g \cup \mathcal{D}_t} \mathcal{L}_c(D_d(F(x_i)), d_i) \tag{8}$$

N is the total number of generated and target samples. d_i is the domain label, where $d_i = 0$ if $x_i \in \mathcal{D}_g$ and $d_i = 1$ if $x_i \in \mathcal{D}_t$. $\mathcal{L}_c$ is the normal cross-entropy loss.

Classification Loss ($\mathcal{L}_{cls}$): The adaptive classifier is trained using the classification loss of generated samples. We update this classifier's parameters based on the loss gradient. The gradient of this loss is also used to train feature extractor to generate class discriminative features.

$$\mathcal{L}_{cls} = \frac{1}{N_g} \sum_{g_i \in \mathcal{D}_g} \mathcal{L}_c(C(F(g_i)), y_i) \tag{9}$$

Here C is the classifier network. N_g are the total number of generated samples.

Total Loss: The total loss is given as below

$$\mathcal{L}(G, F, D_d) = \delta\mathcal{L}_{lik} + \alpha * \mathcal{L}^g_{adv} + \beta * \mathcal{L}_{crs} + \lambda * \mathcal{L}_{dis} + \mu * \mathcal{L}_{cls} \tag{10}$$

where $\delta, \alpha, \beta, \lambda$ and μ are the tuning parameters. In our experiments, α and β are set to 1 and exponentially decreased to 0 while μ is kept 0 until 25 epochs, and later it is set to 1. λ is the adaptation parameter. It is set to 1 throughout the experiments. we set $\delta = 0.1$ in all the experiments. We also optimize the parameters of the adversarial discriminator by minimizing the loss defined in Eq. 6 for a given generator's parameters.

5. Results and Discussion

5.1. Datasets

MNIST→ MNIST-M: We experiment with the MNIST dataset [27] as source data. In order to obtain the target domain (MNIST-M) we blend digits from the original set over patches randomly extracted from color photos from BSDS500 [3]. Due to this, a domain gap is observed, and performance is poor on the MNIST-M classifier. There are 60k samples used to train the MNIST classifier, and 59k samples of MNIST-M are used for adaptation. For adaptation results are shown in Table 1.

SVHN→ MNIST: In this adaptation task, source data (SVHN [45]) and target data (MNIST) both have ten-classes. In this setting, we are provided the classifier trained on the SVHN and unlabeled MNIST dataset. The provided classifier is trained on the full SVHN dataset, and we adapt the full MNIST dataset. There are 60k samples present in the MNIST dataset, while SVHN has 73K samples. The results are reported in Table 1.

MNIST→ SVHN: For the MNIST-SVHN transfer task, the provided pre-trained classifier trained on the MNIST dataset. We use the full SVHN dataset. The classifier is also trained on the full MNIST dataset. The results are reported in Table 1.

MNIST→ USPS: The USPS contains 16x16 grey images. We resized them to 32x32. In this experiment, we use full MNIST and USPS images as the target set. The results are reported in Table 1.

Office-31 [50]: It contains three domains Amazon (A), Webcam (W), and DSLR (D). Each domain has 31 object classes, and we evaluate all the six adaptation task. We obtain the features from ResNet-50 [16], pre-trained on Imagenet.

Source Data Required	*Method*	*MNIST→MNIST-M*	*SVHN→MNIST*	*MNIST→SVHN*	*MNIST→USPS*
Yes	DANN [12]	81.5	71.1	35.7	89.1
	CMD [73]	85.5	86.5	-	
	kNN-Ad [56]	86.7	78.8	40.3	
	DRCN [13]	-	82.0	40.1	91.8
	PixelDA[5]	98.2	-	-	
	ADDA [61]	-	76.0	-	
	ATN [51]	94.2	86.2	52.8	-
	MCD[52]	-	96.2	-	
	JDDA[6]	88.4	94.2	-	
	UDA [10]	99.5	99.3	89.2	
	3CATN [31]	-	98.3	-	96.1
No	Baseline	59.4	67.2	37.7	82.5
	SDDA(ours)	**85.5**	75.5	42.2	**89.9**
	SDDA-P(ours)	84.1	**76.3**	**43.6**	88.5

Table 1: Classification accuracy (%) comparisons with baseline and other state-of-the-art methods on standard digit dataset using the proposed method. Note that the proposed models do not use the source samples for adaptation, while all state-of-the-art methods access the source data. The baseline is without the adaptation method. SDDA-P is referred to when we initialized the classifier with the weight of a pre-trained classifier.

Dataset	***Performance***
MNIST→MNIST-M	87.5
SVHN→MNIST	97.8
MNIST→SVHN	84.6
MNIST→USPS	95.3

Table 2: Classification performance on source data after the adaptation.

5.2. Performance Evaluation

Table 1 shows the results for different adaptation tasks for the proposed method. In the table, baseline refers to the case when there is no adaptation performed. This is one of the pioneering efforts to solve domain adaptation without accessing source data to the best of our knowledge. The SDDA-P is referred to when the classifier is initialized with the pre-trained classifier weight, while SDDA is when it is initialized randomly. Note that all the previous state-of-the-art methods work when source data is accessible. Table 1 shows that the proposed model performs comparably to the baselines that make use of full source information. Table 2 shows the classifier's performance on source dataset after the adaptation. In target data, we achieve a boost in the performance from the baselines; for example, from MNIST →MNIST-M adaptation task, we obtained $\sim 25\%$ improvement. For the other adaptation task, we also obtained improvement with a large margin. In Table 4, the results for the other variant in the training method, we call it adaptation after the generation (SDDA-G), are presented for the MNIST→MNIST-M and MNIST→SVHN adaptation tasks. In this method, we first learn the generative model, and after that, samples are generated to train the adaptation module. The number in the bracket indicates the number of generated samples used for the adaptation. We can observe that initially, the performance improves when we increase the number for generated samples, but later it slightly deteriorates.

For Office-31 dataset, the adaptation results on all the six tasks are reported in Table 3. In the dataset adaptation, we generate the features of corresponding images from the generator. We can observe that we can achieve the $\sim 3\%$ and ~ 1.5 % improvement over the baseline without accessing the source dataset on hard adaptation task A→W and A→D. We implement on Torch-Lua framework.

6. Analysis

6.1. Ablation study on Loss Functions

In Table 5, we show the ablation study of different loss functions used by the proposed model for the MNIST →MNIST-M adaptation task. We can observe that by introducing the likelihood-based loss, we get better improvement. The generative adversarial loss is very crucial to incorporate; the model does not converge without it. The reason is that the generator can not be trained without any adversarial discriminator.

6.2. Ablation on other Domain Adaptation models

This section provides the results for different domain adaptation methods such as MMD [62], IDDA [26], Wasserstein DA [57] and GRL [12]. In these experiments, we use DCGAN architecture for both generator and classifier. This analysis reveals that the proposed method can be plugged into any domain adaptation framework. Results are shown in Table 6 for the MNIST→MNIST-M adaptation task

Source Data Required	*Method*	*A→W*	*D→W*	*W→D*	*A→D*	*D→A*	*W→A*	*Avg*
Yes	DANN [12]	81.2	98.0	99.8	83.3	66.8	66.1	82.5
	GTA [53]	89.5	97.9	97.9	87.7	72.8	71.4	86.5
	DADA [59]	92.3	99.2	100.	93.9	74.4	74.2	89.0
No	Baseline	79.9	96.8	99.5	84.1	64.5	66.4	81.9
	SDDA(ours)	**82.5**	**99.0**	**99.8**	**85.3**	**66.4**	**67.7**	**83.5**

Table 3: Classification accuracy (%) comparisons with baseline and other state-of-the-art methods on Office-31 [50] dataset using proposed method. Note that, the proposed models do not use source samples for adaptation, while all other methods utilize the source data for adaptation.

Method	*MNIST→MNIST-M*	*MNIST→SVHN*
Source Only(0 samples)	59.4	37.7
SDDA-G(300 samples))	64.3	38.5
SDDA-G(2000 samples))	61.8	**39.6**
SDDA-G(6000 samples))	**70.5**	38.8
SDDA-G(40000 samples))	68.7	39.2
Oracle	82.5	39.8

Table 4: Classification accuracies for MNIST→MNIST-M and MNIST→SVHN transfer task for different generated samples for *Adaptation after Generation* variant. Our model is SDDA-G with the number in bracket indicating the number of generated samples used for adaptation. Oracle refer, when actual source data is used for adaptation.

$\mathcal{L}_{lik}$	$\mathcal{L}_{dis}$	$\mathcal{L}^{g}_{adv}$	$\mathcal{L}_{crs}$	$\mathcal{L}_{cls}$	**Accuracy**
-	-	√	√	√	80.6
-	√	√	√	√	83.1
√	√	-	√	√	not converged
√	√	√	√	√	**85.5**

Table 5: Ablation study of different loss functions for the MNIST→MNIST-M adaptation task.

6.3. Distribution Discrepancy

The domain adaptation theory [4] suggests $\mathcal{A}$-distance as a measure of a cross-domain discrepancy, which, together with the source risk, bounds the target risk. The proxy $\mathcal{A}$-distance is defined similar to [47] as $d_{\mathcal{A}} = 2(1-2\epsilon)$, where ϵ is the generalization error of a classifier (e.g. kernel SVM) trained on the binary task of discriminating source and target. Figure 3 shows $d_{\mathcal{A}}$ for MNIST→ MNIST-M adaptation task between source-target, source-generated,and target-generated domains in before adaption and after the adaptation. We can infer from the figure that source and generated domains are always closer. The target domain is closer after the adaptation as compare to before adaptation model.

6.4. Performance on number of Generated Samples

We experiment on the number of generated samples required for the adaptation. In this setting, we first generate the samples without the adaptation module. Both generation and adaptation modules are trained separately. We have experimented with this variant in the MNIST→ MNIST-M and MNIST → SVHN adaptation tasks with different numbers of generated samples. The performance is reported in Table 4. From the table, we can observe that we obtain the best performance MNIST→ MNIST-M when the number of generated samples is selected 6000.

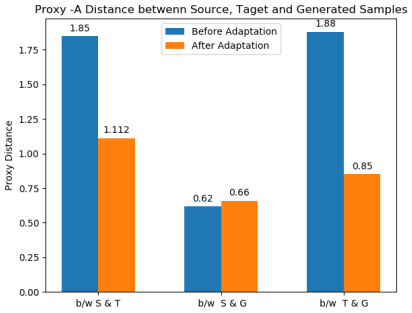

Figure 3: Proxy distance between source-target (S&T), source-generated (S&G) and target-generated (T&G) domains before and after the adaptation for MNIST→ MNIST-M adaptation task. In figure, S, T, and G stands for source data, target data, and generated data respectively.

Method	**Source Only**	**Full Source data**	**Source free(Proposed)**
MMD [62]	59.1	64.3	62.5
IDDA [26]	59.1	82.3	83.0
WDA [57]	59.1	82.8	79.5
GRL [12]	59.1	82.5	**85.5**

Table 6: Performance of different domain adaptation model on MNIST→MNIST-M adaptation task. Source only: when there is no-adaptation, Full source data: full source data is used for adaptation, the Proposed method: samples are generated from the trained classifier, and for adaptation, these dummy samples are used.

λ	0.1	0.3	0.5	0.8	1	1.5	2
SDDA-G(6k)	66.3	66.5	68.7	**70.8**	70.5	66.9	-
SDDA-P	78.9	80.2	84.0	84.0	**84.1**	79.3	77.8
SDDA	82.8	83.4	83.5	83.2	**85.5**	82.6	82.3

Table 7: Ablation study of adaptation parameter λ for MNIST→MNIST-M adaptation task.

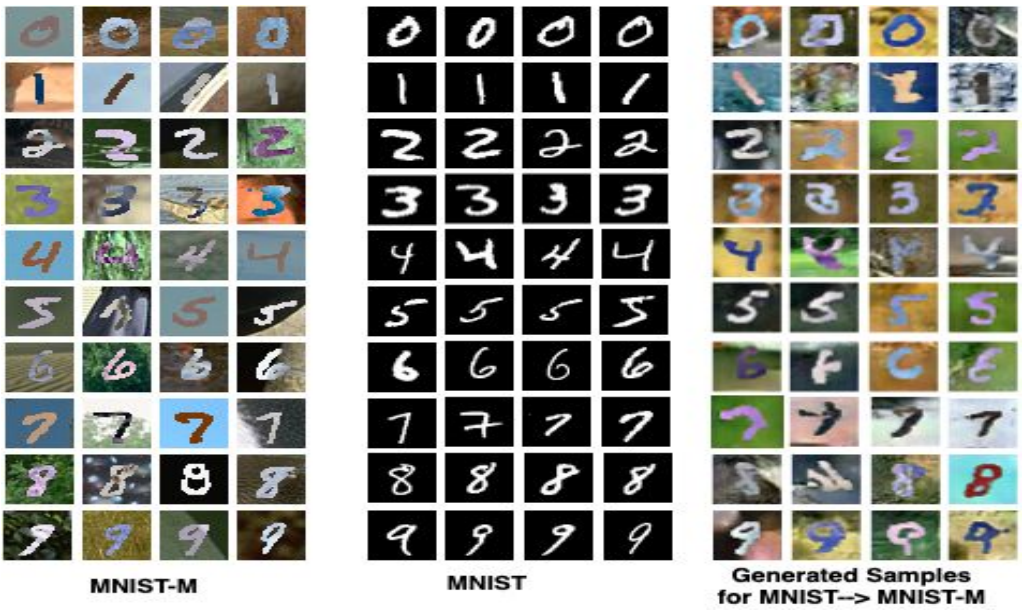

Figure 4: Visualization of source data (MNIST), Target data (MNIST-M)and Generated data for class digit 0-9. We can observe that generated images have proper class discrimination.

	No Adapt		Adapt	
	around Src	around Tgt	around Src	around Tgt
Density	0.771	0.769	0.734	0.776

Table 8: Density estimation generated samples around the source (**Src**) and target (**Tgt**) domains for MNIST→MNIST-M adaptation task on adapted and non-adapted features.

6.5. Image Generation Visualization

In Figure 4, we provide visualization of generated imaged during the adaptation process for MNINT→MNIST-M adaptation task. Here MNIST data is source data, and MNIST-M data is target data. We can observe that the generated images look like the target samples. The reason is that we use the target samples as real data for training the GAN. We can also observe that the generated images are class discriminative, i.e., each sample has one class. This implies that the cross-entropy loss from the pre-trained classifier helps generator to provide the class structure so it can avoid the mode collapsed problem. The third observation is that all the examples are diverse so that we can generate sufficient distinct examples to train the classifier.

6.6. Ablation Study with Adaptation parameter λ

We provide ablation of proposed method for value of λ in Table 7 for MNIST→MNIST-M adaptation. It can be observed in the adaption; the proposed model is not very sensitive to the adaptation value. Performance is better when we choose $\lambda = 1$ for the SDDA model.

.

6.7. Density Estimation of Generated samples

The objective of density estimation is to estimate the closeness of generated samples with source and target domains. We estimate density in both cases i.e around the source data and the target data [19]. For obtaining it, the features are obtained by forward images till convolution layers. We analyze the density estimation using both adapted and non-adapted features. The generated samples density around the source domain is the average number of samples, which can be found within a ϵ neighborhood of source samples. These results are reported in Table 8. This density estimation shows that the generated samples have a similar density with source and target data using the non-adapted model for features. It shows that the distribution of generated samples is equally close to both the source and target dataset. However, in adapted features, the generated samples' density is slightly higher around the target domain.

7. Conclusion

We propose a source data-free adaptation method that solves one of the critical challenges that existing domain adaptation techniques face, i.e., the availability of source data. The proposed approach is generic, i.e., it can be applied with any existing domain adaptation models. The proposed work is one of the novel attempt that tackles the domain adaptation problems without the source data's availability. From the results obtained, we believe that the proposed model provides an exciting avenue for further research on this problem.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Sravanti Addepalli, Gaurav Kumar Nayak, Anirban Chakraborty, and Venkatesh Babu Radhakrishnan. Degan: Data-enriching gan for retrieving representative samples from a trained classifier. In *AAAI*, pages 3130–3137, 2020.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5), 2010.

[4] 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), 2010.

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *(CVPR)*, volume 1, page 7, 2017.

[6] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3296–3303, 2019.

[7] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3514–3522, 2019.

[8] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *AAAI*, pages 3521–3528, 2020.

[9] Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 451–460, 2016.

[10] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1416–1425, 2019.

[11] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[13] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1994–2003, 2018.

[18] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[19] Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. U-dada: Unsupervised deep action domain adaptation. In *Asian Conference on Computer Vision*, pages 444–459. Springer, 2018.

[20] My Kieu, Andrew D Bagdanov, Marco Bertini, and Alberto Del Bimbo. Domain adaptation for privacy-preserving pedestrian detection in thermal imagery. In *International Conference on Image Analysis and Processing*, pages 203–213. Springer, 2019.

[21] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018.

[22] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[23] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC, 2018.

[24] Vinod Kumar Kurmi, Vipul Bajaj, Venkatesh K Subramanian, and Vinay P Namboodiri. Curriculum based dropout discriminator for domain adaptation. *BMVC*, 2019.

[25] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P Namboodiri. Attending to discriminative certainty for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–500, 2019.

[26] Vinod Kumar Kurmi and Vinay P Namboodiri. Looking back at labels: A class based domain adaptation technique. In *International Joint Conference on Neural Networks (IJCNN)*, July 2019.

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

[28] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[29] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019.

[30] Nam LeTien, Amaury Habrard, and Marc Sebban. Differentially private optimal transport: application to domain adaptation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2852–2858. AAAI Press, 2019.

[31] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Zi Huang. Cycle-consistent conditional adversarial transfer networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 747–755, 2019.

[32] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[33] Shuang Li, Chi Harold Liu, Binhui Xie, Limin Su, Zhengming Ding, and Gao Huang. Joint adversarial domain adaptation. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 729–737, New York, NY, USA, 2019. Association for Computing Machinery.

[34] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *ICML*, 2020.

[35] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, pages 4013–4022, 2019.

[36] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

[38] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[39] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2188–2196, 2018.

[40] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016.

[41] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018.

[42] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751, 2019.

[43] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. In *International conference on articulated motion and deformable objects*, pages 85–94. Springer, 2018.

[44] Arun Reddy Nelakurthi, Ross Maciejewski, and Jingrui He. Source free domain adaptation using an off-the-shelf classifier. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 140–145. IEEE, 2018.

[45] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[46] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[47] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[48] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2020.

[49] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

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

[51] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org, 2017.

[52] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[53] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[54] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*, pages 1262–1273, 2019.

[55] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2178, 2019.

[56] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

[57] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, 2018.

[58] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[59] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *AAAI*, pages 5940–5947, 2020.

[60] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[61] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

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

[63] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation.

[64] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, volume 33, pages 5345–5352, 2019.

[65] Jun Wen, Nenggan Zheng, Junsong Yuan, Zhefeng Gong, and Changyou Chen. Bayesian uncertainty matching for unsupervised domain adaptation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 3849–3855. AAAI Press, 2019.

[66] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018.

[67] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[68] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2281, 2017.

[69] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[70] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[71] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[72] Liuchun Yuan, Congcong Ruan, Haifeng Hu, and Dihu Chen. Image inpainting based on patch-gans. *IEEE Access*, 7:46411–46421, 2019.

[73] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. 2016.

[74] Chenrui Zhang and Yuxin Peng. Visual data synthesis via gan for zero-shot video classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1128–1134, 2018.

[75] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[76] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018.