# Adversarial Domain Adaptive Autoencoders

**Prateek Munjal, Sujit Rai**
Indian Institute of Technology Ropar

## Abstract

We propose a novel approach of reducing the shift between target and source domain using the image to image translation technique. We achieve this image to image translation by learning an autoencoder with a GAN discriminator. The autoencoder is responsible for translating a data point in the source domain to a data point in target domain. The enocder of autoencoder is viewed as a domain invariant feature extractor whereas the decoder is viewed as a GAN generator. In a similar vein, the discriminator is responsible to introduce the adversarial loss. In contrast to traditional adversarial game setup between the generator and discriminator, we set up the adversarial game between autoencoder and discriminator. We investigate our proposed model, ADA-AE, with both quantitative and qualitative experiments where we show that the decoder/generator do learns the marginal distribution of target distribution. Thus we observe an improvement in accuracy over test domain.

## 1 Introduction

Recent advances in deep neural networks have produced state the of the art results in various machine learning tasks. To a considerable extent, the success of the neural network is appreciated by the the amount of labeled data available. One may note that the labeled data is not always available in enormous amount either because of the dynamic changes in environment or it is a very expensive procedure. However, there are some cases when the labeled (source)data is available, but this data suffers from a shift when compared to actual target data distribution. This labeled data can be obtained synthetically or semi-synthetically via some automatic procedure or a program. Therefore, to infer the properties in target domain, we define the task of transferring the knowledge from the source domain (labeled synthetic data) to the target domain (real world data) which is formally known as domain adaptation.

Depending on the type of target domain, the domain adaptation can be categorized as supervised domain adaptation and unsupervised domain adaptation. For supervised domain adaptation, the target domain consists of labeled data and in contrast, unsupervised domain adaptation considers only the data points in target domain. Since we propose a method for unsupervised domain adaptation therefore the paper will focus on unsupervised setting only. The unsupervised domain adaptation can be further classified depending on the type of classes present in the source and target domains. Partial domain adaptation is defined when there may exist a partial overlap between the labels of the source and target domains, i.e., the source and target domain might consist of **few or no common** labels. In contrast to partial domain adaptation, the complete domain adaptation assumes a complete overlap in labels between the source and target domains. The task of domain adaptation can be further classified depending on the complexity of the covariate shift i.e whether there always exists a single optimal classifier performing well on both the source and target domain. When such a classifier exists, we term it the task as conservative domain adaptation, otherwise non-conservative domain adaptation.

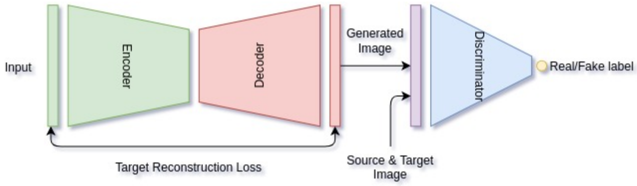

Figure 1: Proposed method using Image-to-Image translation

In our problem setting, we assume that the inter-domain labels are completely overlapping and there exist a classifier such that it performs well on both the domains. Therefore in this paper we focus on solving the task of unsupervised and conservative domain adaptation. The approaches which are extensively used in domain adaptation literature are Adversarial Methods [Wen *et al.*, 2018] [Wang *et al.*, 2018] [Chadha and Andreopoulos, 2018] [Laradji and Babanezhad, 2018], Network Methods [Chen *et al.*, 2018] [Mancini *et al.*, 2018], Optimal Transport [Damodaran *et al.*, 2018]. Our proposed method uses the pseudo labeling approach with adversarial methods to reduce the shift. We exploit traditional autoencoders to learn a transformation from a data point in source domain to a data point in target domain. In contrast to GANs [Goodfellow *et al.*, 2014], the decoder (generator) receives the input from encoder distribution. We hypothesize that the encoder can act as an invariant feature extractor which used by the decoder network can transform a source data point to target data point, thereby reducing the shift between domains. We also view our decoder as a GAN generator producing the data points in target domain. We enlist the main contributions of the paper which are as follows

(i) We propose a novel method using autoencoders to learn a function which translates a data point in source domain to a data point in target domain.

(ii) We leverage this learned translating function to assign the pseudo-labels to the transformed data points in target domain.

(iii) Empirically we also observe that using the image to image translation, the decoder (generator) of our proposed model implicitly learns the marginal distribution of target data.

The paper is structured as follows. We start with first introducing the related work in section 2 which is followed by our proposed methodology. We then discuss our experiments in section 3 where we show both our qualitative and quantitative results and then end with a conclusion and future work in section 4 and 5 respectively.

## 2 Related Work

In this section, various previous work which can be helpful in designing the deep architecture for domain adaptation are discussed. The theory behind domain adaptation was proposed by [Ben-David *et al.*, 2010] and [Ben-David *et al.*, 2007]. [Ben-David *et al.*, 2010] proposed the theory that bounds the expected error on the target domain by using three terms. 1) Expected error in the source domain $R_S(h)$. 2) The discrepancy between two classifiers that are used to classify source and target domains $\frac{1}{2}d_{H\Delta H}(S,T)$. 3) The shared expected error of a single classifier on both source and target domain $\lambda$.

$$\forall h \in H, R_T(h) \leq R_S(h) + \frac{1}{2}d_{H\Delta H}(S,T) + \lambda \qquad (1)$$

Here, $R_T(h)$ is expected error on target domain.

Another theory [Ben-David *et al.*, 2007] bounds the error on the target domain by using the three terms 1) expected error on the source domain. 2) H-distance for divergence in the domain of source and target distribution. 3) shared error on both domains.

$$\forall h \in H, R_T(h) \leq R_S(h) + \frac{1}{2}d_H(S,T) + \lambda \qquad (2)$$

Here, $\frac{1}{2}d_H(S,T)$ is the H-distance. In case, of conservative domain adaptation, the shared error $\lambda$ is very low since there exists a single classifier that can classify both source and target domains. Therefore, $\lambda$ term is negligible and is neglected. While in case of non-conservative domain adaptation $\lambda$ term is high and can't be neglected.

Most of the work focus on reducing this H-distance by minimizing the covariate shift between the two distributions using moment maching such as MMD, CMD [Tzeng *et al.*, 2014] [Long *et al.*, 2015] [Long *et al.*, 2016b] [Long *et al.*, 2016a], adversarial loss [Tzeng *et al.*, 2017] [Ganin and Lempitsky, 2014] [Bousmalis *et al.*, 2017] or batch normalization statistics [Li *et al.*, 2016]. One of the disadvantage of this approach is the requirement of the use of kernel which can be considered as a type of hyperparameter, Since deep architectures are known to extract underlying features automatically therefore, it is preferred to avoid the use of hand crafted features as much as possible. Another approaches for domain adaptation focus on using adversarial loss in order to reduce the domain discrepancy.

**Unsupervised Domain Adaptation using BackPropagation** [Ganin and Lempitsky, 2014]. This was the most earlier work which used adversarial training for domain adaptation. In this work they used a single feature extractor which would extract discriminative and domain invariant features from the input. The output of this feature extractor was then passed to the class classifier. A domain discriminator was used in order to force the feature extractor to extract domain invariant features. The job of domain discriminator was to predict the domain of the output of the feature extractor, while the job of feature extractor was to fool the domain discriminator and minimize the classification loss. The intuition behind this paper formed the basis of lot of work on domain adaptation that used adversarial loss. Some of the work that used adversarial loss for domain adaptation are discussed below.

**Importance Weighted Adversarial Nets for Partial Domain Adaptation** [Zhang *et al.*, 2018]. This work was an extension of Partial Transfer Learning with Selective Adversarial Networks [Cao *et al.*, 2017] which used k discriminator if there were k classes thus it was infeasible to use this approach if there were large number of classes. This work overcame the need for k discriminators by using only 2 discriminators. Since the task was to perform partial domain adaptation therefore, it was not possible to directly match the marginal distribution since the conditional distribution of labels were different. Therefore this approach matched the marginals of only the instances which belonged to the labels which were also present in target domain. In-order to identify the instances which belonged to the common class as that of target domain, a domain discriminator was used. The basic intuition behind this approach is that if an instance does not belong to the common class then the discriminator will be able to easily predict the domain of the instance. Thus if domain classifier is very confident for a instance then it means that the instance does not belong to the common class. This domain classifier was then used to provide a lower weight to the instances which were not present in the common class and higher weight to the instances which were present in the common class. The resulting weighted instances were then used for training the feature extractor and another domain discriminator in order to align the weighted marginal distributions.

**Cons :** This work focused only on aligning the marginal distributions while it was assumed that the conditional distribution will get aligned automatically once marginals are aligned. Also the assumption that the instances that belong to the same class won't get classified easily by the domain discriminator won't hold good in all the cases if there is lot of variation in the probability distribution of both domains.

**Cycle-Consistent Adversarial Domain Adaptation** [Hoffman *et al.*, 2017]. Most of the previous work in domain adaptation focused only on aligning marginal distributions and it was assumed that the conditional distribution of labels will get aligned automatically however this assumption does not hold good in all cases. Also prior works did not enforce any constraint on maintaining semantic information while translating from source to target. Therefore this work focused on overcoming this issue. They used image to image translation in order translate an instance from source domain to target domain using cycle consistency. Thus maintaining one to one mapping and aligning marginals in the pixel space. Inorder to align conditional distribution of labels, Source classifier was used as a noisy labeller to ensure that the translated image in the target domain gets classified in the same way as that of the original image in the source domain. This work used adversarial loss for aligning marginals in the feature space as well as in the pixel space and source classifier as noisy labeller for aligning the conditional distributions.

**Cons :** Using source classifier as the noisy labeller inorder to maintain the semantic consistency and aligning the conditional distributions is a weak constraint. Also the image to image translation performed in this method is a one to one translation while in many cases every single image from the source domain can get mapped to many different images in the target domain.

Following Approaches are quite different in the sense that they do not use adversarial loss for domain adaptation.

**Dirt-T approach for Domain Adaptation** [Shu *et al.*, 2018]. This work is the most recent work in the field of domain adaptation and the procedure discussed in this work can be applied with any existing model for domain adaptation. This work focused on overcoming two main issues present in most of the adversarial approaches for domain adaptation. 1.) If the feature extractor function has a very high capacity and if the source-target supports are disjoint then enforcing domain-invariance in the feature space is a very weak constraint. 2.) If the task of domain adaptation is non-conservative then there doesn't exist a single classifier that will perform better in case of both source and target domains. The critical component of this paper is the cluster assumption, which states that decision boundaries should not cross high-density regions. Conditional cross entropy was used for maintaining cluster assumption. Minimization of conditional entropy forces the classifier to be confident on the unlabeled target data. Inorder to enforce locally lipschitz constraint virtual adversarial training was used. This technique was named as Virtual Adversarial Domain Adaptation (VADA). The paper further extended this approach by proposing decision boundary iterative refinement training approach (DIRT-T) for non conservative domain adaptation. This approach first creates a decision boundary in the source domain using the VADA approach then it iteratively refines the decision boundary on target domain by taking a seemingly small step that minimizes the conditional entropy subject to the constraint that the KL Divergence between previous prediction and current prediction is small. The above mentioned pros of Dirt-T can therefore, be useful for partial domain adaptation.

**Learning To Cluster In Order To Transfer Across Domains And Tasks** [Hsu *et al.*, 2017]. This work is very different from the conventional approach towards domain adaptation. The approach discussed in this section can be used for domain adaptation tasks as well as for transferring knowledge across tasks. The approach is to transform the problem into problem of finding the pairwise similarity between instances of a particular domain and then learn a transferable similarity function that can be used for clustering in both domains. After similarity function is learned it can be used to form the clusters in both the target and source domains and then the class labels will be assigned to a cluster on basis of the majority of instances from a particular label present in the cluster of a particular domain.

## 3 Methodology

**Notations** We use $P_s(x)$ and $P_t(x)$ for denoting the source distribution and target distribution in some $\mathcal{X}$ space. We use the notations $\mathbf{x}_s$ and $\mathbf{x}_t$ for denoting a data point in source domain and target domain respectively. On input $\mathbf{x}$, we represent the output of the encoder and the discriminator network as $z = Enc(\mathbf{x})$ and $y = Dis(\mathbf{x})$ respectively. The output $y \in \{0, 1\}$ denotes the probability of classifying that whether an input $\mathbf{x}$ belongs to target distribution ($y = 1$) or not ($y = 0$). Similarly on input $\mathbf{z}$, the output of decoder network is denoted by $\tilde{\mathbf{x}} = Dec(\mathbf{z})$. Thus if $\mathbf{x}$ is some data point in source or target domain, $\mathbf{x}_s$ or $\mathbf{x}_t$, then $\tilde{\mathbf{x}}$ is $\tilde{\mathbf{x}}_s$ or $\tilde{\mathbf{x}}_t$.

Our proposed method **A**dversarial **D**omain **A**daptive **A**utoencoders, which we term as ADA-AE, combines the traditional autoencoders with GANs [Goodfellow *et al.*, 2014]. We view the encoder as a feature extractor such that it extracts the inter-domain invariant features. Similarly the decoder is viewed as a GAN generator which translates a latent code, $\mathbf{z}$, containing invariant features to a data point, $\tilde{\mathbf{x}}$, in the target domain. Therefore combined autoencoder learns a function which when given an input data point in source domain, it translates to a data point in target domain. We further incorporate the adversarial learning by setting up an adversarial game between autoencoder and discriminator. For visualizing the proposed approach, we refer to Figure 1. The autoencoder network tries to synthesize the samples which seems plausible to discriminator whereas the discriminator tries to discriminate the generated samples (fake) and samples from target distribution (real). Since we use an autoencoder with discriminator network, we have three set of parameters in total. We define the autoencoder parameters as $\{\theta_{enc}, \theta_{dec}\}$ and discriminator parameters as $\theta_{disc}$.

Now, we formally define the losses for training the ADA-AE model. The encoder parameters ($\theta_{enc}$) and decoder parameters ($\theta_{dec}$) are jointly trained by minimizing the translation loss ($\mathcal{L}_{trans}$). The translation loss consist of two terms, *i.e* , reconstruction loss ($\mathcal{L}_{recons}$) and gan loss ($\mathcal{L}_{gan}$). The ($\mathcal{L}_{recons}$) is used to preserve the tight coupling between encoder and decoder network whereas the $\mathcal{L}_{gan}$ incorporates the GAN generator like properties. Thus $\mathcal{L}_{trans}$ is defined as follows

$$\mathcal{L}_{trans} = \alpha \mathcal{L}_{recons} + \beta \mathcal{L}_{gan} \tag{3}$$

where,

$$\mathcal{L}_{recons} = \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2^2 \tag{4}$$

$$\mathcal{L}_{gan} = -\log(\text{Dis}(\tilde{\mathbf{x}})) \tag{5}$$

and $\tilde{\mathbf{x}} = Dec(Enc(x)), \forall x \in \{x_s \cup x_t\}$. The $\alpha$ and $\beta$ are the hyper parameters which are learned empirically.

In contrast to previous approaches where we used to transfer the knowledge of source domain to target domain, we deliberately limit the use of $\mathcal{L}_{recons}$ on target data points and hypothesize that we can transfer the knowledge from target domain to source domain via autoencoder. We observe that this knowledge of transfer would help the autoencoder learn the translation function better. We achieve this transferring of knowledge in target domain by using the same autoencoder, thereby the same parameters, to reconstruct the source images in target domain. We empirically show that without explicitly matching the source labels while reconstruction for source data points, we still get the accurate (in terms of labels) reconstructions of source in target domain. For example, in context of MNIST handwritten digits, if we pass an image of class 3 to the autoencoder, then it outputs a generated image of 3 in target domain. Thus we leverage this implicit property learned in autoencoder by assigning a pseudo label, *i.e* source image label, to the transformed source image in target domain. The loss incurred by the discriminator,$\mathcal{L}_{disc}$, is formulated as a standard discriminator loss with a slight modification. The standard discriminator loss contains two terms, where one term corresponds to loss for real sample and the other term for loss of fake sample. Besides these two terms, we add a third term where we explicitly mention the source data points as fake samples to the discriminator. We speculate that the gradients corresponding to third term, coming from the discriminator to the generator (decoder) will avoid the generator to produce an image in source domain. Thus the decoder is restricted to construct the image in target domain only, which is our desired goal. We formally define the $\mathcal{L}_{disc}$ as follows

$$\begin{aligned}\mathcal{L}_{disc} = -\big[\log(\text{Dis}(\mathbf{x}_t)) + \log(1 - \text{Dis}(\mathbf{x}_s))\\ + \log(1 - \text{Dis}(\tilde{\mathbf{x}}))\big], \forall x \in \{x_s \cup x_t\}\end{aligned} \tag{6}$$

From a training perspective, we refer to algorithm 1 for training ADE-AE model. The line 14 in our algorithm shows how we leverage the pseudo labeling power of autoencoder.

# 4 Experiments

We investigate our proposed method by performing both qualitative and quantitative experiments on real world benchmark dataset MNIST [LeCun *et al.*, 1998]. We use MNIST as source distribution whereas to get fair results we construct the target distribution to be considerably complex, i.e, MNIST-M [Ganin *et al.*, 2016]. For qualitative analysis, we visualize the latent features of ADA-AE autoencoder and in the context of quantitative analysis, we compare the accuracy on a held out target dataset.

**Algorithm 1** ADE-AE Training Schedule

```
1: $\theta_{enc}, \theta_{dec}, \theta_{disc} \leftarrow$ Initialize parameters
2: $x_s, y_s \leftarrow$ sample minibatch from source with labels
3: $x_t \leftarrow$ sample minibatch from target domain
4: $X \leftarrow \{x_s \cup x_t\}$
5: $Z \leftarrow \text{Enc}(X)$
6: $\tilde{X} \leftarrow \text{Dec}(Z)$
7: while not convergence do
8:     $\theta_{enc} + \theta_{dec} \xleftarrow{+} -\nabla_{(\theta_{enc}+\theta_{dec})}(\mathcal{L}_{trans})$
9:     $\theta_{disc} \xleftarrow{+} -\nabla_{\theta_{disc}}(\mathcal{L}_{disc})$
10: end while
11: $z_s \leftarrow \text{Enc}(x_s)$
12: $\tilde{x}_t \leftarrow \text{Dec}(z_s)$
13: Assign $y_s$ labels to the transformed $\tilde{x}_t$ data points
14: $X' \leftarrow \{\{x_s, y_s\} \cup \{\tilde{x}_t, y_s\}\}$
15: Train a classifier C, on $X'$
16: Check target accuracy on C.
```

## 4.1 Qualitative Analysis

To visually analyze the shift between source and target distribution, we analyze the latent features of autoencoder. For non-adaptive method, we construct an autoencoder which is trained for reconstructing both the source and target data points by minimizing the standard mean squared loss between the input and reconstructed data point. Similarly to analyze the extent of adaptiveness learned by our ADA-AE model, we visualize the TSNE plot of latent codes of autoencoder of ADA-AE trained using algorithm 1. For a fair comparison, the architecture of both adaptive and non-adaptive autoencoders is kept same, so that both falls under the same model complexity, hence the same power. From figure 2, it is very evident that the ADA-AE autoencoder learns to reduce the shift while in the non-adaptive case, the shift is clearly observed.

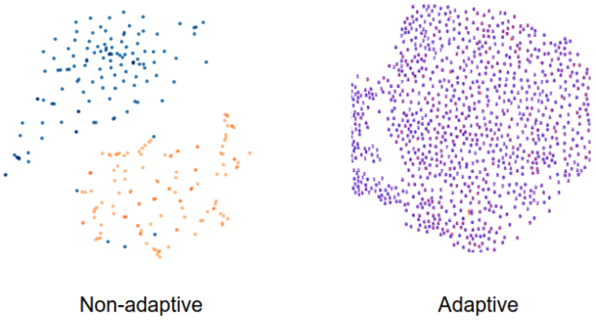

Figure 2: T-SNE visualization of source and target embeddings in non-adaptive (left) and adaptive (right) autoencoder.

Another set of experiments were performed to verify the class labels of target images generated. As observed from figure 3, the generated target images which are present on the right has the same class as that of the source images. Thus we can say that there happens to be a transfer of class discriminative features from the source domain to the target domain.

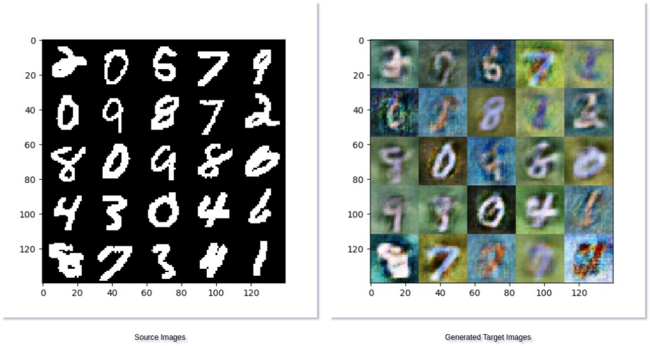

Figure 3: (Left) Original Source images. (Right) Generated Target Images

We further visualized the reconstructed target images to check whether the encoder was learning domain invariant features and decoder was learning to generate the target images from this invariant representation. As can be seen from the figure 4, the original MNIST-M target images are on the left and the corresponding reconstructed target images on the right.

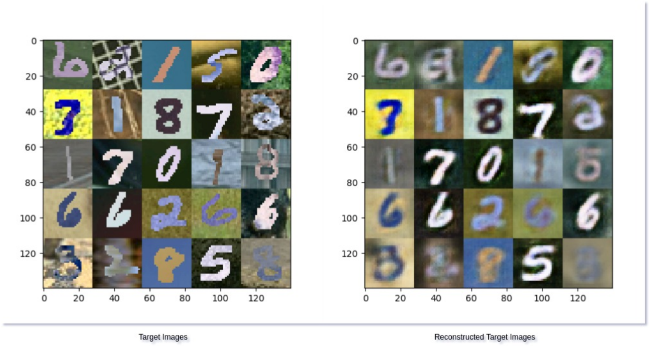

Figure 4: (Left) MNIST-M samples. (Right) Reconstructed Target Images

### 4.2 Quantitative Analysis

To analyze the improvement due to pseudo labelling of target samples, we compute the accuracy obtained over the held out target set on a classifier which is trained on the dataset consisting of labelled source images and pseudo labelled generated target images. We observe that when a classifier was trained on only labelled source images and then tested on unseen target images, the accuracy on target comes around 46%. Similarly when the classifier of same model complexity was trained on both labelled source images and pseudo labelled target images then the accuracy on test target images was 52%. Such increase in accuracy becomes significant since we compute the accuracy over large set of instances, precisely, 5k. Moreover, the improvement in accuracy supports our hypothesis that using the autoencoder for translating images from source domain to target domain, helps in transferring the knowledge between domains and thus reduce the covariance shift.

| Approach | Target Accuracy(%) |
|---|---|
| Source Only | 46 |
| ADE-AE | 52 |

Table 1: Accuracy comparison on test data.

## 5 Conclusion

From both the quantitative and qualitative results obtained, we hypothesize that the use of image to image translation approach in domain adaptation seems a promising research direction. Since after observing the Figure 2, it is quite evident that the generated target data points had a lower covariate shift with respect to the source data points, thus we see an improvement in accuracy too. As we observe in figure 3, that the generated images in target domain preserves the class of source data points, therefore the use of reconstruction loss (refer Equation 6) on the target data points do leads to matching of conditional distributions of generated target images and input source image to a considerable extent.

## 6 Future Work

One can extend the current model by replacing the Autoencoder by a Variational autoencoder. The variational autoencoder, VAE, can be leveraged to generate new instances of target domain, thus increasing the data for classification task. Other contrasting approach could be the use of wasserstein distcriminator instead of standard GAN discriminator, thus exploring the learning behaviour of the ADA-AE model under 1 lipshitz constraint. It is quite evident from the qualitative analysis of latent representation where it was observed that the conditionals did not match but the marginal do. Further, an auxiliary classifier can be used in the discriminator which in addition to real/fake probability also predicts the class labels which can help in matching the conditionals or some other distance metric can be used for reducing the distance between instances belonging to same class and increases the distances between instances of different class.

## References

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[Ben-David *et al.*, 2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[Bousmalis *et al.*, 2017] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference* *on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[Cao *et al.*, 2017] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. *arXiv preprint arXiv:1707.07901*, 2017.

[Chadha and Andreopoulos, 2018] Aaron Chadha and Yiannis Andreopoulos. Improving adversarial discriminative domain adaptation. *arXiv preprint arXiv:1809.03625*, 2018.

[Chen *et al.*, 2018] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. *arXiv preprint arXiv:1808.09347*, 2018.

[Damodaran *et al.*, 2018] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. *arXiv preprint arXiv:1803.10081*, 2018.

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

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

[Goodfellow *et al.*, 2014] 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.

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

[Hsu *et al.*, 2017] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2017.

[Laradji and Babanezhad, 2018] Issam Laradji and Reza Babanezhad. M-adda: Unsupervised domain adaptation with deep metric learning. *arXiv preprint arXiv:1807.02552*, 2018.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324, 1998.

[Li *et al.*, 2016] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

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

[Long *et al.*, 2016a] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

[Long *et al.*, 2016b] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[Mancini *et al.*, 2018] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. *arXiv preprint arXiv:1805.01386*, 2018.

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

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

[Tzeng *et al.*, 2017] 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.

[Wang *et al.*, 2018] Jiawei Wang, Zhaoshui He, Chengjian Feng, Zhouping Zhu, Qinzhuang Lin, Jun Lv, and Shengli Xie. Domain confusion with self ensembling for unsupervised adaptation. *arXiv preprint arXiv:1810.04472*, 2018.

[Wen *et al.*, 2018] Jun Wen, Risheng Liu, Nenggan Zheng, Qian Zheng, Zhefeng Gong, and Junsong Yuan. Exploiting local feature patterns for unsupervised domain adaptation. *arXiv preprint arXiv:1811.05042*, 2018.

[Zhang *et al.*, 2018] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8156–8164, 2018.