Unsupervised Domain Adaptation using Generative Models and Self-ensembling

Eman T. Hassan  
Indiana University Bloomington  
emhassan@indiana.edu

Xin Chen  
Midea ETC  
chen1.xin@midea.com

David Crandall  
Indiana University Bloomington  
djcran@indiana.edu

Abstract

Transferring knowledge across different datasets is an important approach to successfully train deep models with a small-scale target dataset or when few labeled instances are available. In this paper, we aim at developing a model that can generalize across multiple domain shifts, so that this model can adapt from a single source to multiple targets. This can be achieved by randomizing the generation of the data of various styles to mitigate the domain mismatch. First, we present a new adaptation to the CycleGAN model to produce stochastic style transfer between two image batches of different domains. Second, we enhance the classifier performance by using a self-ensembling technique with a teacher and student model to train on both original and generated data. Finally, we present experimental results on three datasets Office-31, Office-Home, and Visual Domain adaptation. The results suggest that self-ensembling is better than simple data augmentation with the newly generated data and a single model trained this way can have the best performance across all different transfer tasks.

1. Introduction

Large scale annotated dataset and increasing computational power have enabled the rapid development of deep convolutional networks (CNNs) that produce high performance on many computer vision problems like classification, segmentation, and detection [24, 30, 41]. Unlike human beings who easily generalize knowledge across different domains, it is not easy for CNN models to generalize across other datasets having different characteristics. As a result, domain adaptation, which aims at enhancing the ability of the network to generalize across different domains, has emerged as a hot topic in recent years.

Training a generative model to generate images with different characteristics has witnessed significant progress. Especially after the introduction of Generative Adversarial Networks (GAN) [16]. GANs enable the development of deep models which are capable of image generation across different domains with high quality [39, 20, 23]. Therefore, GANs have been widely applied to many areas such as image inpainting [36], super-resolution [27], pixel-level domain adaptation [4], and image-to-image translation [54, 18].

That has inspired us to use GANs to randomize the generation of multiple instances of the dataset of different styles to accommodate for any possible unseen domain shifts. We then use these newly generated datasets to effectively enhance knowledge transfer in a teacher-student framework. Previous work [22, 5] has examined using generative networks as a part of the overall adaptation between two domains, while in our case we examine the possibility of generative networks to generate many random instances of the original datasets, where each instance can represent a specific domain shift. Finally we employ the newly obtained datasets to enhance zero-shot domain adaptation.

In this paper, we propose a model that can generalize across different domain shifts. This is accomplished in two steps, the overall proposed system is shown in Figure 1. First, we introduce stochastic style transfer as an adaptation to the CycleGAN network [54]. It transforms the function of the module from performing one-to-one image translation, to instead perform stochastic style transfer, which generates images with adapted style between the source and target domains. The adaptation depends on relaxing some of the mapping constraints between the two domains. It only checks the mapping between the two domains based on a disentangled representation of the images into style and content representations. Second, we use the trained modules to generate newly adapted datasets. Then, to achieve zero-shot domain adaptation, we develop self-ensembling based domain adaptation, which adapts the teacher-student architecture proposed in [8], where different instances of the same data are the source of perturbation. The experimental results suggest that the newly proposed architecture produces better performance on transferring between one domain to many others.

In summary, we make the following major contributions in this paper:

1. Showing that GAN networks can help make a model
Figure 1: The overall system architecture. The input source and target images are sampled randomly from Office-31 [43], Office-Home [50], and Visual Domain adaptation [38] datasets. The images are fed to the CycleGAN-based Stochastic Style Transfer module (explained in more detail in Figure 2). Then style adapted images are generated for both source and target domains. All the images, source and target (both original and adapted), are fed to a teacher-student DA classifier system (explained in more detail in Figure 3).

2. Using CycleGAN-based architecture to perform stochastic style transfer between the two domains, which helps generate multiple instance of the original datasets.

3. Employing a self-ensembling technique to train a model both in supervised and unsupervised ways using the original datasets and the newly generated instances for both source and target images.

4. Showing that the self-ensembling architecture is better in training than fine tuning with the newly generated data.

2. Related Work

Domain adaptation is a machine learning problem whose goal is that a model trained on a source data $D_s(I_s, Y_s)$ can generalize to related but different data $D_t(I_t, Y_t)$. The source data is fully labelled while the target data is either partially labeled (semi-supervised case) or totally unlabeled (unsupervised case). With this paradigm, many both shallow and deep techniques were proposed to tackle this problem. There are examples of Shallow Techniques like Sup-space alignment [9], second order statistic alignment known as CORAL [45], the employment of landmarks to enhance feature alignment [1, 15, 3], sample reweighting [19], and Metric based learning as in [53, 7, 25]. Some examples of Deep domain adaptation are knowledge distillation with soft-targets [17] and selectively choosing data samples for fine-tuning [12]. Generally, deep techniques can be categorized into discrepancy based methods, reconstruction based techniques and adversarial based techniques [6].

Discrepancy based methods as in [31, 32, 51, 46] aim at alignment the deep feature embedding between the source and target domains by optimizing a loss function that penalizes distribution mismatch. Long et al. [31] proposed a domain adaptation network that uses a maximum mean discrepancy loss function to match the distribution of the fully connected layers of the source and target networks, while Long et al. [32] extended this work by using a joint maximum mean discrepancy loss function instead, and Yan et al. [51] incorporated the class weights in the loss function by employing an expectation maximization training paradigm. Similar to CORAL [45] in shallow techniques, Sun et al. [46] used Deep CORAL to match the second order statistics between the deep features of the two networks.

Adversarial based techniques [10, 29, 49, 4] use adversarial settings to make the deep feature embeddings of both domains similar. Ganin et al. [10] introduced gradient reverse units that reverse the gradient of the domain classifier so that the network learns similar feature representations for both domains. Tzeng et al. [48] incorporated loss functions like domain confusion and domain classification in the network framework. Tzeng et al. [49] proposed a general framework for unsupervised domain adaptation in both cases of generative or discriminative mappings. Bousmalis et al. [4] introduced a technique that adapts source-domain images to appear as if drawn from the target domain. Reconstruction based techniques outline how to construct a latent representation that is shared across multiple domains. Ghifary et al. [13] considered inter-domain variances as sources of noise in a denoising encoder, then make it learn common features across these different domains. Similarly Ghifary et al. [14] proposed a deep reconstruction network...
that learns both labeled image prediction and the reconstruction of both target and source images. Bousmalis et al. [5] proposed a separation network that learns to separate the latent space into two components, one private to each domain and the other common across different domains. Peng et al. [37] proposed a deep generative alignment network that reconstructs images from the source domain that are similar to the target domain and trains the classifier based on these generated images. In our work, we assume a zero-shot framework where we penalize the difference on prediction of the target data, without actually trying to match the distribution of the source domain.

**Image-to-Image Translation** techniques describe the generation of an image that represents a translation of the input image from one domain to another [21, 54, 18, 55, 22, 35, 28]. Liu et al. [28] proposed a UNIT framework for image translation that is based on generative adversarial networks and variational autoencoders that creates a shared latent space which can generate corresponding images between the two domains. Zhu et al. [54] presented the CycleGAN model which introduced cycle consistency loss to construct a mapping between the two domains, but this model constructs a one-to-one mapping between the two domain. This work was followed by many papers that introduces models that make the translation many-to-many, as in [55] where Zhu et al. introduced a hybrid model that combines conditional variational autoencoder-GANs and Conditional Latent Regressor-GANs. Kang et al. [22] used an image generation model to enhance domain adaptation and made the source network guide the attention alignment of the target network in an expectation maximization framework. In our work we extended CycleGAN to generate multiple style adaptations to the original domain to help in knowledge transfer.

**Semi-supervised learning and domain adaptation using teacher student model.** Zagoruyko et al. [52] showed that the performance of a student network is enhanced when it mimics the attention mechanism of the teacher network. Kang et al. [22] has employed deep adversarial attention between the two networks to achieve domain adaptation between the two domains. Mean-teacher model [47] and self-ensembling [26] were proposed in the framework of semi-supervised learning. Laine et al. [26] employed the consensus in prediction of the network in training to enhance network performance. While Tarvainen et al. [47] proposed the mean teacher model to aggregate information after each step not epoch, in this model the teacher weights used exponential moving average weights of the student model. French et al. [8] extended these ideas to the problem of domain adaptation. Meanwhile Luo et al. [33] consider the connection between different data points to build a graph that employs a self-ensemble based on smooth neighbors on teacher graphs.

### 3. Algorithm

#### 3.1. CycleGAN based stochastic Style Transfer

The motivation of this adaptation is how to generate images that have the same contents of the source domain but randomized style that can match in some aspect different domain shifts. The CycleGAN module [54] constructs a mapping between the two domains $X, Y$ as $F : X \rightarrow Y$, $G : Y \rightarrow X$ where the cycle consistency aims at producing $X \approx G(F(X))$. That module is designed to produce a one-to-one mapping between the two domains $X, Y$. We propose a modification to the architecture so that the image generated does not belong to either $X$ or $Y$, but contains a shared style representation across the two domains. This architecture is shown in Figure 2. The architecture assumes we can obtain a disentangled representation for the images into both content and style representations, so that each domain generator can produce an image that matches the content representation of the original domain and matches random style representation of the other domain. In this way we can benefit from the module to generate multiple random dataset instances that represent different shifts with respect to the original domain.

To get a disentangled feature representation for image $I$ into content $I_c$ and style $I_s$ representation, we employ the method in [11]. An input image $I_s$ is passed through a pre-trained Vgg-16 model[44] where $I_s = \text{relu}_{12}(I)$ and $I_t$ is the concatenation of $Gm(\text{relu}_{12}(I))$, $Gm(\text{relu}_{22}(I))$, $Gm(\text{relu}_{33}(I))$, $Gm(\text{relu}_{43}(I))$, and $Gm(\text{relu}_{53}(I))$, where $Gm$ represents the gram matrix of the features of $\text{relu}_{ij}(I)$ represents the Rectified Linear Unit activation applied to the output of the $j^{th}$ convolution module in layer $i$.

In the proposed architecture $U_s$ and $U_t$ represent the source and target generator networks respectively which are implemented using the Unet model architecture [42]. Input images $I_s$ and $I_t$ from the source and target domains are forwarded to the system such that $I_{su} = U_s(I_s)$ and $I_{tu} = U_t(I_s)$. Then the content and style representations are extracted for both input and generated images. For example $I_{sc}$ and $I_{st}$ are the content and style representations, respectively, of $I_s$, and similarly $I_{tuc}$ and $I_{tul}$ are the representations for $I_{tu}$. We introduce an Intra-domain loss function $l_{in}$ and cross domain loss function $l_{cross}$ to be able to generate images with similar contents of the original domain but similar style of the other domain. The Intra-domain loss function $l_{in}$ penalizes the difference in the content representation between the input and generated image of the same domain:

$$L_{in} = \text{MSE}(I_{sc}, I_{suc}) + \text{MSE}(I_{tc}, I_{tuc}) \quad (1)$$

while the cross domain loss function $l_{cross}$ penalizes the difference in the style representation across the other do-

\[
L_{cross} = \text{MSE}(I_{sc}, I_{tuc}) + \text{MSE}(I_{tc}, I_{suc})
\]
main:

\[ L_{\text{cross}} = MSE(I_{sl}, I_{sul}) + MSE(I_{tl}, I_{tul}) \]  \hspace{1cm} (2)

Then we have the cycle loss represented by \( l_{\text{recons}} \). The reconstruction loss \( l_{\text{recons}} \) contributes to training convergence and generating realistic images, and is computed by

\[ L_{\text{rec}} = MSE(I_s, I_{srec}) + MSE(I_t, I_{trec}) \]

where, \( I_{srec} = U_t(U_s(I_s)) \) \hspace{1cm} (3)

where, \( I_{trec} = U_s(U_t(I_t)) \)

Finally we calculate the adversarial loss functions \( l_{\text{adv1}}, l_{\text{adv2}} \) and \( l_{\text{adv3}} \), respectively. Adversarial losses \( l_{\text{adv1}}, l_{\text{adv2}} \) are employed so that the discriminator networks \( D_{\text{style}_s}, D_{\text{style}_t} \) confuse the domain generator networks \( U_s \) and \( U_t \) to generate images with style similar to the style of the other domain.

\[ l_{\text{adv1}} = \min_{\{U_s, U_t\}} \max_{\{I_s, I_t\}} \{E_{I_s}[\log(D_{\text{style}_s}(I_s))]+E_{I_t}[\log(1-D_{\text{style}_t}(U_t(I_t)))]\} \hspace{1cm} (4) \]

\[ l_{\text{adv2}} = \min_{\{U_s, U_t\}} \max_{\{I_s, I_t\}} \{E_{I_s}[\log(D_{\text{style}_s}(I_s))]+E_{I_t}[\log(1-D_{\text{style}_t}(U_s(I_s)))]\} \hspace{1cm} (5) \]

On the other hand \( l_{\text{adv3}} \) and \( D_{\text{content}} \) confuse the generator networks \( U_s \) and \( U_t \) to generate realistic images.

\[ l_{\text{adv3}} = \min_{\{U_s, U_t\}} \max_{\{I_s, I_t\}} \{E_{\{I_s, I_t\}}[\log(D_{\text{content}}(\{I_s, I_t\}))]+E_{\{I_s, I_t\}}[1-\log(D_{\text{content}}(\{U_s(I_s), U_t(I_t)\}))]\} \hspace{1cm} (6) \]

The final loss function is a weighted average of these losses, and is computed as:

\[ l_{\text{total}} = \lambda_1 \times l_{\text{in}} + \lambda_2 \times l_{\text{cross}} + \lambda_3 \times l_{\text{rec}} + \lambda_4 \times l_{\text{adv1}} + \lambda_5 \times l_{\text{adv2}} + \lambda_6 \times l_{\text{adv3}} \hspace{1cm} (7) \]

3.2. Self-ensemble zero shot domain adaptation

The motivation of this section is that simple data augmentation may not be the best way to train a classifier benefiting from the generated images. As shown in Figure 1 that we have both labelled source images and labelled randomly adapted source images and unlabelled target images and unlabelled randomly adapted target images, and data augmentation doesn’t explore the dependencies that exists between each image and the corresponding adapted instance.

The idea of a network to have consistent behaviour under different perturbations has been used to enhance the performance of the trained networks \([2, 40, 34]\). In our work we
explore the idea that the perturbation can be adding a random new style to the image as shown in Figure 3. In this architecture we extend the framework provided by [8] but we employ the perturbation to be stochastic style transfer between the two domains instead of random data augmentation.

In this model the student is trained in a supervised way using the source dataset instances (the original one Data$_s$ and the style mapped one Data$_{sm}$). The student model is also trained in an unsupervised manner which is related to the difference in prediction between the student network and the teacher network to the original target data Data$_t$ and style mapped target Data$_{tm}$. On the other hand, the teacher network model is updated in an exponential moving average manner with the student network. The supervised loss is described by:

$$L_{sup}(\theta_{Student}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} \{1\{y^{(i)} = j\} \times \log \left( \frac{e^{\theta_{Student}^{T} x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_{Student}^{T} x^{(l)}}} \right) \}$$

The unsupervised loss is described by:

$$L_{unsup} = N_{Student}(I_t) - N_{Teacher}(I_{tm})$$

4. Experimental Results

We conducted experiments on three datasets for an image classification task:

1. **Office-31** dataset [45]. It consists of 31 categories and contains 4,110 images distributed over three domains: 1) Amazon domain which contains 2,817 images that are collected from Amazon.com, 2) Webcam domain containing 795 web camera images, and 3) DSLR domain containing 498 images from SLR cameras. In our work we evaluate the transfer of $A \rightarrow \{W, D\}$, $W \rightarrow \{A, D\}$ and $D \rightarrow \{W, A\}$.

2. **Office-Home** dataset [50]. It consists of 65 categories in 15,590 images over four domains: 1) **Art** domain with 2427 images, 2) Clipart domain of 4,365 images downloaded from multiple clipart websites, 3) Product domain of 4,439 images gathered from Amazon.com, and 4) Real-world domain of 4,357 images. Art and Real-World domains were built from websites like www.deviantart.com and www.flickr.com. In our work we evaluate the transfer task $Ar \rightarrow \{Cl, P, R\}$, $Cl \rightarrow \{Ar, P, R\}$, $P \rightarrow \{Cl, Ar, R\}$ and $R \rightarrow \{Cl, P, Ar\}$.

3. **Visual Domain Adaptation** classification task (VisDa) [38]. It consists of over 280K images across 12 categories across two domains: 1) Synthetic and 2) Real. In our work we evaluate the transfer between Synthetic $\rightarrow$ Real.

**CycleGAN-based Stochastic Style Transfer.** The results of the Stochastic Style Transfer generation module are shown in Tables 1, 2, and 3, where Table 3 shows the results across multiple transfer tasks in all three datasets, while Tables 2 and 3 show results across different training epochs in the same transfer task across different training epochs. These results are shown in a Table format where each cell in the table contains two blocks of images. Each block consists of three rows, the first row represents the original images $I_s$ or $I_t$ for source and target domains respectively (source on the left, target on the right). The second row represents the output of applying the generator network to the input image denoted as $I_{su}$ or $I_{tu}$ for source and target images respectively. The third row represents the cycle reconstructed images $I_{srec}$ or $I_{trec}$.

For training the module we used a batch size of 8, and we used the whole dataset in case of **Office-31**, but in the case of **Office-Home** we use 85% and in case of **VisDa** source we used 0.5% and 5% for the target. We did not sample the overall dataset in the cases of **Office-Home** and **VisDa** in order to reduce training time due to very small batch size in training. We choose $\lambda_i = 1$ in Eq 7 for $i = 1 \ldots 6$.

The results show that the module manages to create different image instances of the original domain that have style similar to the images of the other domain. Table 1 shows the transfer results for the Office-Home dataset between two domains $R \rightarrow Ar$ across different training epochs. It illustrates that the system managed to produce photo realistic results across many epochs except for epoch 25, where the results are little bit washed out. This demonstrates that with different statuses of the module parameters $U_s$ and $U_t$ we can obtain different style transfer results. The figures show that the module managed to keep the contents of the
original domain but transfer the style of the other domain in a batch based mode. The results for the VisDa dataset are shown in Table 2. The performance is less compelling compared to the other dataset as it is a more challenging task. Still the system managed to produce good transfer results between the two domain as in Epoch 10 and Epoch 25. Finally results across all transfer tasks for all the datasets in Epoch 5 of the training algorithm are shown in Table 3. These results illustrate that the system managed to provide good results except for some cases as in Office-Home (Art → Product) and VisDa (Synthetic → Real).

For each domain’s data we can generate many different instances of that data using the generation modules trained for different transfer tasks. For example consider the Amazon domain in the Office-31 dataset, which has its original version $D_A$ and the version generated due to generation module (Amazon → DSLR) known as $D_{A,A,D}$ or due to the module (Amazon → Web) known as $D_{A,A,W}$. In each generation module we can choose training parameters stored from different training epochs (in our case we used epoch 5) and we can choose to get the images generated to be $(I_{su}, I_{tu})$ or the reconstructed images $(I_{src}, I_{trec})$ (in our case we used $(I_{su}, I_{tu})$).

**Self-ensemble zero shot domain adaptation.** The classification results are shown in Tables 4b, 4a, and 4c for datasets Office-31, Office-Home, and VisDa datasets, re-
Table 3: The generation results of epoch 5 across the three datasets and the different transfer tasks. Each cell represents the result of one transfer task.

Table 4d. In our experiments we trained three different types of classification models with color codes: 1) Base model $M_{base}$: a model that has been trained on original source images only in a supervised way, the source datasets have color described by $Data_{T1}$. 2) Tuned model $M_{tune}$: for a model that has been trained in a supervised way using the original source images and the source mapped images (simple data augmentation to the source), the source datasets both have color described by $Data_{T1}$. 3) $M_{ensemble}$ model: is a model trained in a supervised way using the data described by the source datasets (with color $Data_{T1}$) while trained in the unsupervised way using the target datasets with color described by $Data_{T3}$. Finally, all these models have been tested against other datasets that have not been in-
(a) Top-5 classification results for Office-Home dataset, showing that model M 11 achieves the best performance across all other transfer tasks.

| Dataset | M 0 | M 1 | M 2 | M 3 | M 4 | M 5 | M 6 | M 7 | M 8 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| D_{Ar} | 0.859 | 0.4983 | 0.4972 | 0.5985 | 0.7908 | 0.6044 | 0.8260 | 0.5127 | 0.5076 |
| D_{Ar,Cli} | 0.3072 | 0.2065 | 0.1979 | 0.2312 | 0.5355 | 0.2560 | 0.3701 | 0.2527 | 0.2496 |
| D_{Ar,R} | 0.0515 | 0.2784 | 0.2879 | 0.3288 | 0.5424 | 0.3920 | 0.6045 | 0.3381 | 0.3299 |
| D_{Ar,p} | 0.4693 | 0.2676 | 0.2907 | 0.3374 | 0.5186 | 0.4093 | 0.6589 | 0.3434 | 0.3380 |
| D_{Cl} | 0.5092 | 0.4609 | 0.4836 | 0.5320 | 0.4706 | 0.5023 | 0.4919 | 0.7924 | 0.7905 |
| D_{Cl,Cli} | 0.3306 | 0.4695 | 0.2895 | 0.3262 | 0.3770 | 0.3542 | 0.3698 | 0.5546 | 0.5896 |
| D_{Cl,R} | 0.3259 | 0.4427 | 0.2967 | 0.3264 | 0.3816 | 0.3718 | 0.3907 | 0.6598 | 0.5674 |
| D_{Cl,R} | 0.3469 | 0.4836 | 0.3139 | 0.3328 | 0.3967 | 0.3771 | 0.3914 | 0.5685 | 0.5685 |
| D_{R} | 0.6261 | 0.5785 | 0.9074 | 0.7276 | 0.5815 | 0.7103 | 0.6227 | 0.6012 | 0.6128 |
| D_{R} | 0.3173 | 0.2529 | 0.4147 | 0.3112 | 0.3110 | 0.3587 | 0.3501 | 0.3522 | 0.3106 |
| D_{P} | 0.4256 | 0.3695 | 0.5605 | 0.4486 | 0.4218 | 0.5035 | 0.4424 | 0.4291 | 0.4238 |
| D_{P,R} | 0.4105 | 0.4229 | 0.6851 | 0.3363 | 0.4638 | 0.5785 | 0.5029 | 0.4745 | 0.4885 |
| D_{R} | 0.7113 | 0.6296 | 0.6998 | 0.8964 | 0.6688 | 0.8737 | 0.7020 | 0.6472 | 0.6403 |
| D_{R} | 0.4773 | 0.3766 | 0.4036 | 0.5354 | 0.4847 | 0.7461 | 0.5348 | 0.4476 | 0.4399 |
| D_{R} | 0.2990 | 0.2679 | 0.2821 | 0.3278 | 0.3584 | 0.3876 | 0.3398 | 0.3168 | 0.3197 |
| D_{R} | 0.5084 | 0.4024 | 0.4574 | 0.6071 | 0.4970 | 0.6863 | 0.5355 | 0.4745 | 0.4638 |

(b) Top-5 classification results for Office-31 dataset, showing that model M 8 achieves the best performance across all other transfer tasks.干活 in training, either supervised or unsupervised, with color described by Data_{T4}.

Results Analysis. The performance results show that there is a model that has the highest performance on all other target domains. Table 4b presents the results of the Office-31 dataset showing that Model M 8 has the highest performance across all the other transfer tasks. This model has been trained using the proposed method with supervised source data D_D and D_{D,W,D} while trained on D_W and D_{W,D,W} in an unsupervised way. Despite that this model has never seen any instance of domain A, it has the highest transfer performance across all instances of domain A. Similarly, the results of the Office-Home dataset are shown in Table 4a. From this table, we can see that model M 11 has the highest performance over all other target datasets except one case of data D_P. This model was trained on D_R and D_{R_AR,R} in a supervised way while it was trained on D_{Ar} and D_{Ar,Ar,R} in an unsupervised way, and managed to generalize across other domains of P, Cl without seeing these data. These results are consistent with Top-1 results. Finally, the results of VisDA are presented in Table 4c, showing that model M 1 achieves the best transfer top-5 results, while on top-1, M 2 and M 3 achieve best results on D_{Re} and D_{Re,Re}, respectively. No single model managed to get higher performance, we believe this is due to less quality of Generated random instances shown in Table 2.

5. Conclusion and Future Work

In this work, we showed that domain adaptation can benefit from generative models to enhance network generalization performance across multiple other domains. We demonstrated that these networks can produce multiple random instances of the same domain dataset. Each instance represents a random different shift compared to the original dataset. Our results also show that using a self-ensemble method is better than simple data augmentation to enhance knowledge transfer performance even on unseen domains. In the future we plan to train the model end-to-end where each epoch of the generation module produces data with new random domain shifts and train the classifier incrementally, so that the model can generalize across continuous domain shift changes. Moreover, we plan to conduct experiments in which we compare the highest performing model across different methods in different transfer tasks in the same dataset, which we did not do here because our goal was to show that a model can generalize even across unseen domains by using the generative models, which to our knowledge has not been done before.
References

[1] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 56–63, 2015. 2
[2] P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. In Advances in Neural Information Processing Systems, pages 3365–3373, 2014. 4
[3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and G. Csurka. Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374, 2017. 2
[4] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In Proceedings of the 24th international conference on Machine learning, pages 209–216. ACM, 2007. 2
[5] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In ICLR, 2018. 1, 3, 5
[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1):2096–2030, 2016. 2
[7] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In Advances in neural information processing systems, pages 601–608, 2007. 2
[8] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1866–1875. IEEE, 2017. 1
[9] G. Kang, L. Zheng, Y. Yan, and Y. Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. arXiv preprint arXiv:1801.10068, 2018. 1, 3
[10] K. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In ICLR, 2018. 1
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 1
[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4681–4690, 2017. 1
[13] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In Advances in Neural Information Processing Systems, pages 700–708, 2017. 3
[14] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In Advances in Neural Information Processing Systems, pages 469–477, 2016. 2
[15] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In International Conference on Machine Learning, pages 222–230, 2013. 2
[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014. 1
[17] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. stat, 1050:9, 2015. 2
[18] E. Hosseini-Asl, Y. Zhou, C. Xiong, and R. Socher. Augmented cyclic adversarial learning for domain adaptation. arXiv preprint arXiv:1807.00374, 2018. 1, 3
[19] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In Advances in neural information processing systems, pages 601–608, 2007. 2
[20] A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4681–4690, 2017. 1
ceedings of the 32nd International Conference on Machine Learning (ICML), 2015. 2

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), 2017. 2

[33] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In IEEE International Conference on Computer Vision, 2018. 3

[34] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 2018. 4

[35] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In IEEE International Conference on Computer Vision, 2017. 3

[36] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2536–2544, 2016. 1

[37] X. Peng and K. Saenko. Synthetic to real adaptation with generative correlation alignment networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1982–1991. IEEE, 2018. 3

[38] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924, 2017. 2, 5

[39] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2015. 1

[40] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In Advances in Neural Information Processing Systems, pages 3546–3554, 2015. 4

[41] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 1

[42] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015. 3

[43] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In European conference on computer vision, pages 213–226. Springer, 2010. 2, 4, 5

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Proc. International Conference on Learning Representations, 2015. 3

[45] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In Thirtieth AAAI Conference on Artificial Intelligence, 2016. 2

[46] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In European Conference on Computer Vision, pages 443–450. Springer, 2016. 2

[47] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in neural information processing systems, pages 1195–1204, 2017. 3

[48] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In Proceedings of the IEEE International Conference on Computer Vision, pages 4068–4076, 2015. 2

[49] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2962–2971. IEEE, 2017. 2

[50] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 5385–5394. IEEE, 2017. 2, 5

[51] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 945–954. IEEE, 2017. 2

[52] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In ICLR, 2017. 3

[53] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua. Robust distance metric learning with auxiliary knowledge. In IJCAI, pages 1327–1332, 2009. 2

[54] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In IEEE International Conference on Computer Vision, 2017. 1, 3

[55] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In Advances in Neural Information Processing Systems, pages 465–476, 2017. 3