Duplex Generative Adversarial Network for Unsupervised Domain Adaptation

Lanqing Hu[1,2] Meina Kan[1,3] Shiguang Shan[1,3] Xilin Chen[1]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology
lanqing.hu@vipl.ict.ac.cn {kanmeina,sgshan,xlchen}@ict.ac.cn

Abstract

Domain adaptation attempts to transfer the knowledge obtained from the source domain to the target domain, i.e., the domain where the testing data are. The main challenge lies in the distribution discrepancy between source and target domain. Most existing works endeavor to learn domain invariant representation usually by minimizing a distribution distance, e.g., MMD and the discriminator in the recently proposed generative adversarial network (GAN). Following the similar idea of GAN, this work proposes a novel GAN architecture with duplex adversarial discriminators (referred to as DupGAN), which can achieve domain-invariant representation and domain transformation. Specifically, our proposed network consists of three parts, an encoder, a generator and two discriminators. The encoder embeds samples from both domains into the latent representation, and the generator decodes the latent representation to both source and target domains respectively conditioned on a domain code, i.e., achieves domain transformation. The generator is pitted against duplex discriminators, one for source domain and the other for target, to ensure the reality of domain transformation, the latent representation domain invariant and the category information of it preserved as well. Our proposed work achieves the state-of-the-art performance on unsupervised domain adaptation of digit classification and object recognition.

1. Introduction

The deep learning technique has achieved great success in many area including the computer vision [26, 45, 22, 16, 5], speech recognition [23, 7, 21, 41, 49], etc. Generally, the deep models are usually trained on a large scale labeled training data and tested on the data which share similar distribution as the training one. Otherwise, the performance degenerates badly when the distribution of the training and testing data are different. So for a new task, the training data are usually needed to recollect and that is quite expensive, even impossible. Transfer learning is a preferable technique that can alleviate the cost of recollecting large scale labeled data by transferring knowledge from a different but related sophisticated domain. Domain adaptation is a sub-problem of the general transfer learning, which pays attention to the case where the training data (i.e., source domain) and the testing data (i.e., target domain) share the same task but follow different distributions [37], as shown in Figure 1.

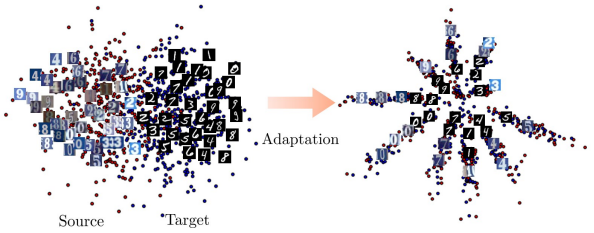


Figure 1: Illustration of domain adaptation. Domain adaptation attempts to mitigate the discrepancy between source and target domain. After adaptation, the source and target domains are expected to share the same or similar distribution, i.e., samples of the same class from both source and target domains should be close to each other. The source domain samples are represented by red circles and target samples are blue ones.

According to the degree to which the data of target domain is labeled, domain adaptation can be categorized into supervised, semi-supervised and unsupervised domain adaptation. In supervised domain adaptation scenario [25] and semi-supervised one [9, 27], all or part of the target domain samples are labeled but the number of labeled data is too limited to learn a satisfactory model for the target domain. In unsupervised domain adaptation scenario [1, 12], all target domain samples are unlabeled. For all three scenarios, the source domain samples are labeled. This work mainly focuses on the unsupervised domain adaptation, of which only related works are reviewed below.

In the early days, most methods deal with the unsupervised domain adaptation problem via instance re-weighting

to obtain the shared similar distribution with the target domain, such as sample selection bias [50, 11, 24] and covariate shift [46, 3]. These instance re-weighting approaches are suitable for those scenarios where the source and target domains share the same support.

However, in many wild scenarios, the supports of source and target domains are different which means the instance re-weighting is not applicable. Then methods targeting on extracting domain invariant representation come up. In [8], the label information is propagated across different domains to extract the cross-domain representation via a co-clustering based algorithm. In [36], the Transfer Component Analysis (TCA) tries to learn some transfer components across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD) to minimize the discrepancy of two domains. In the approach of Sampling Geodesic Flow (SGF) [20], each domain is modeled as a point on a Grassmann manifold, and the intermediate subspaces are obtained by sampling points along the geodesic between the two domain subspaces to model the domain shift. This work is further extended as geodesic flow kernel (GFK) [18], which integrates an infinite number of the subspaces to model domain shift between the source and target domain by the GFK. In [17], a set of landmarks, i.e., a subset of labeled data from the source domain that are distributed most similarly to the target domain, are discovered to bridge the source and the target domain. In [43] and [44], both source and target domain data are projected to a common subspace with low-rank constraint to reduce the domain discrepancy.

Encouraged by the deep models developed in recent years, many approaches come up to alleviate the discrepancy between source and target domain through deep feature learning. The methods proposed in [31] and [32] embed deep features into Reproducing Kernel Hilbert spaces (RKHS) and minimize the maximum mean discrepancy (MMD) metric of the features for feature adaptation. In [15], a deep reconstruction-classification network (DRCN) is proposed to learn common representation for both domains through the joint objective of supervised classification of labeled source data and unsupervised reconstruction of unlabeled target data. In [4], the domain separation networks (DSN) extracts feature representations that are partitioned into two components, one for the private information of each domain and the other for the shared representation across domains to reconstruct the images and features from both domains.

In the approaches above, a metric is usually needed in the objective to measure the discrepancy between domains. The most commonly used metrics include MMD, K-L and Bregman divergence [14, 36, 31, 32]. Recently, inspired by the generative adversarial nets (GANs) [19, 34], the adversarial learning strategy is introduced to restrain the domain discrepancy for better domain adaptation [47, 51, 30, 29].

Following the existing GAN-based domain adaption methods, this work proposes a duplex generative adversarial net named DupGAN to achieve domain invariant feature and domain transformation. As shown in Figure 2, the proposed DupGAN consists of an encoder, a generator and duplex discriminators. The encoder embeds input images from both domains to latent representation; the generator decodes the latent representation to source and target domain images conditioned by a domain code to achieve domain transformation; the generated images are expected to look like those real source and target domain images, therefore, the generator is pitted against the following duplex discriminators, one for distinguishing the real or fake source domain images and the other for target domain. In addition, either discriminator is not only responsible for the real/fake discrimination to restrict the images from the generator to be real, but also the categorial classification for real images to enforce the latent representation domain invariant and preserve its category information. To do the final classification, a classifier is established on the latent representation, which can be also used to predict the labels of target domain images further used in the training stage. Our proposed approach achieves quite promising performance on digit classification and object recognition tasks.

Briefly, our contributions lie in three folds: (1) A generative adversarial network with duplex discriminators named DupGAN is proposed to restrict the latent representation domain invariant with its category information preserved and ensure realistic domain transformation; (2) A classifier is stacked on the latent representation for the final classification, which also predicts the labels of target domain images to make the latent representation discriminative and further used in the duplex discriminators; (3) DupGAN achieves the state-of-the-art performance on unsupervised domain adaptation of digit classification and object recognition.

2. Related Work

The works in [12, 13, 48] handle the domain shift by augmenting a gradient reversal layer or employing the adversarial loss for target domain samples. They are adversarial discriminative methods endeavoring to learn discriminative and domain invariant representation. The methods based on generative adversarial networks (GANs) also reduce the distribution discrepancy by generating samples approximating the distribution of source or target domain. In [47], a conditional generative adversarial network maps samples from the target domain to the source one and applies the source domain classifier to the target domain feature space which is aligned to source. In [30, 29], a tuple of GANs, each for one image domain, is designed to map both domain samples into

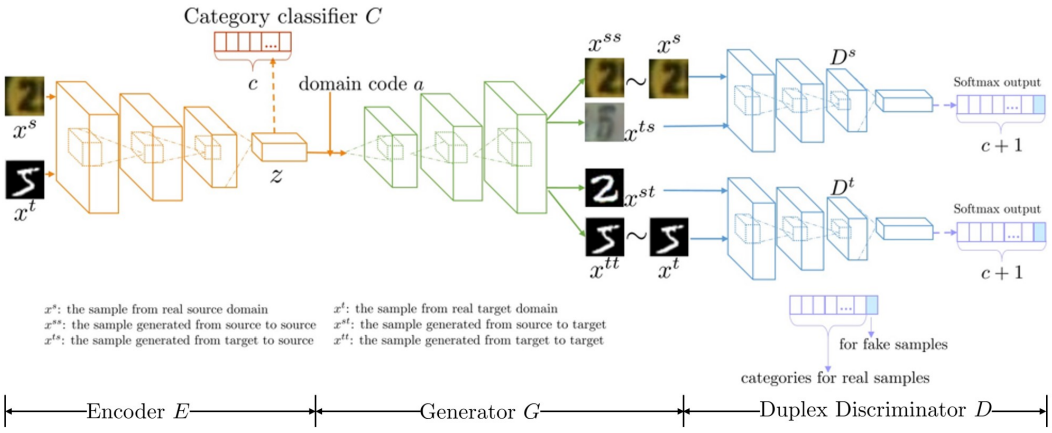


Figure 2: Overall structure of our DupGAN. It composes of three parts, an encoder E, a generator G and duplex discriminators D^s and D^t. The encoder embeds the source domain sample x^s and target domain sample x^t into latent representation z. The generator decodes z into source and target domain images respectively conditioned on a domain code a, i.e., domain transformation of $x^s \xrightarrow{s} x^{ss}$, $x^s \xrightarrow{t} x^{st}$, $x^t \xrightarrow{t} x^{tt}$ and $x^t \xrightarrow{s} x^{ts}$. The transformed images x^{ss} and x^{tt} are expected to be the same as the input images x^s and x^t respectively, forming a self-reconstruction constraint. The duplex discriminators D^s and D^t aiming for distinguishing the generated images x^{ts}/x^{st} from the real images x^s/x^t and categorizing x^s and x^t as well. To do the final classification, a classifier C is built based on the latent representation z, expecting z to preserve both the common feature of source and target domains and category information.

a shared latent space and further decode any latent representation into both source and target domains. CycleGAN [51] proposes a cyclic mapping, i.e., source-target-source and target-source-target sample translation to implement image translation. IcGAN [38] and DR-GAN [33] achieve image translation and domain invariant feature extraction by a conditional generative adversarial net with only one encoder for all domains (not one encoder for only one domain), controlling the domain to be transferred to by a domain indicator code.

Pseudo Labeling

The re-labeling unlabeled sample methods [42, 40] also attempt to deal with domain discrepancy between source and target domain. In [42], the labeling metrics are optimized by utilizing the k-nearest neighbors between the unlabeled target samples and labeled source samples. The method proposed in [40] back-propagates the category loss for the target samples based on pseudo-labeled samples.

3. Method

3.1. Overview

For clear description, we first give some definitions. The labeled source domain images and the unlabeled target domain images are denoted as $X^s = \{(x_i^s, y_i^s)\}_{i=1}^n$ and $X^t = \{x_j^t\}_{j=1}^m$, respectively. The source and target domains share the same c categories but follow different distribution. Unless otherwise specified, the symbols s and t used in the superscript or subscript denotes the source domain and target domain respectively.

Our proposed DupGAN is equipped with one encoder, one generator, and duplex adversarial discriminators aiming for domain invariant feature extraction and domain transformation. An overview of the proposed method is shown in Figure 2. The encoder, denoted as E, attempts to map any image from either source or target domain into a latent representation $z = E(x)$, which is expected to be domain invariant and category informative. The generator, denoted as G, decodes the latent representation into source or target domain images conditioned by a domain code, i.e., achieve domain transformation. The generator is pitted against duplex discriminators, one for source domain and the other for target domain, denoted as D^s and D^t, respectively, to constrain the latent representation input into the generator to be domain invariant and the images from the generator be real. The classifier stacked on the encoder, denoted as C, tries to discriminate the categories of the images from both the source and target domains, and also contributes to the final classification.

Specifically, each discriminator not only discriminates the real images from fake (i.e., generated) images, but also distinguishes the category information of real images, in order to enforce the latent representation z to be domain invariant and with category information preserved. The labels of both source domain images and its counterpart from the generator G are available which can be directly used in D^s and D^t. However, the labels of target domain images and its counterpart from the generator G are unavailable, so the pseudo categorial labels predicted from the classifier C are

used when optimizing D^t.

3.1.1 Encoder and Generator

The encoder E aims for transforming an input image from either source or target domain into a latent representation z as follows:

$$z = E(x), x \in X^s \cup X^t, \tag{1}$$

where E can be any kind of deep neural network with parameters denoted as W^E. For convenience, the latent representation of any source or target domain sample is denoted as $z^s = E(x^s)$ and $z^t = E(x^t)$, respectively. Then, the whole latent representation space of source and target domain images are denoted as $Z^s = E(X^s)$ and $Z^t = E(X^t)$, respectively. So $z \in Z = Z^s \cup Z^t$.

The latent representation z is expected to be domain invariant. First, consider only one path that transforming all samples into only source domain. The generator G tries to decode z into source from either domain. Thus, z tends to be a roughly (not necessarily totally) joint subspace affined to source. Similar for the other path for target, z tends to be a roughly joint subspace affined to target. During the optimization with two paths, the gradients for z from both paths will compromise when they conflict with each other. This will lead to a common subspace, let alone the existence of dominance of commonality which is proved to be the basis of domain adaptation in [2]. The generator G is formulated as below:

$$x^a = G(z, a), z \in Z^s \cup Z^t, \tag{2}$$

where the domain code $a \in \{s, t\}$ is used to specify which domain the latent representation is transformed to. Similar as E, G can be also any kind of deep neural network with parameters denoted as W^G. The input image can be from either source or target domain, and it can be transformed into both the source and target domains respectively. Therefore, the generator G generates four types of images, detailed as follows:

$$x^{ss} = G(z^s, s) = G(E(x^s), s), \tag{3}$$

$$x^{st} = G(z^s, t) = G(E(x^s), t), \tag{4}$$

$$x^{ts} = G(z^t, s) = G(E(x^t), s), \tag{5}$$

$$x^{tt} = G(z^t, t) = G(E(x^t), t). \tag{6}$$

For easy implementation, the latent representation and the domain code is concatenated into one long vector, i.e., $[z; a]$, which is used as input of G. For the source domain image x^s, when transformed to source domain, the generated image x^{ss} should be the same as itself, i.e., $x^{ss} \sim x^s$, and when transformed to the target domain, the generated image x^{st} should look alike real target domain images with category unchanged which is constrained by the discriminator D^t introduced in Section 3.1.2. Similarly, the generated image x^{tt} should be the same as itself, i.e., $x^{tt} \sim x^t$, and x^{ts} should look alike real source domain images with category unchanged which is constrained by the discriminator D^s introduced in Section 3.1.2.

In summary, the objective function of the encoder and generator is formulated as below:

$$\begin{aligned} L_G = & \min_{W^G, W^E} \big(\textstyle\sum_{x^s \in X^s} (H(D^t(x^{st}), \tilde{y}^{st}) + \alpha ||x^{ss} - x^s||_2^2) \\ & + \textstyle\sum_{x^t \in X^t} (H(D^s(x^{ts}), \tilde{y}^{ts}) + \alpha ||x^{tt} - x^t||_2^2) \big) \\ = & \min_{W^G, W^E} \big(\sum_{x^s \in X^s} (H(D^s(G(E(x^s), t)), \tilde{y}^{st}) + \alpha ||G(E(x^s), s) - x^s||_2^2) \\ & + \sum_{x^t \in X^t} (H(D^s(G(E(x^t), s)), \tilde{y}^{ts}) + \alpha ||G(E(x^t), t) - x^t||_2^2) \big). \end{aligned} \tag{7}$$

where $H(\cdot, \cdot)$ is the cross entropy loss used in softmax layer, and α is a balance parameter for the two terms. The 1st and 3rd terms enforce the latent representation z to preserve both cross-domain and category information via the adversarial learning between the generator G and the duplex discriminators D^s and D^t. The 2nd and 4th terms are the reconstruction constraint for those images generated from their original domain.

The sample x^{st} is transformed from the labeled source domain sample x^s, so its category is expected to be the same as x^s's. Its label for learning the generator is re-formulated by including a real/fake discrimination node as follows:

$$\tilde{y}^{st} = [\overbrace{\underbrace{0, 0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{c-i}}^{y^s}, 0], x^{st} \in X^{st}, cat(x^{st}) = i, \tag{8}$$

where $cat(\cdot)$ indicates the category of sample x^{st}, and the y^s with c nodes is the one-hot categorial coding of x^s. The sample x^{ts} is transformed from the labeled source domain sample x^t, so its category is expected to be the same as x^t's. However, x^t is unlabeled before, therefore, the categorial label of x^{ts}, which is also the label of x^t, needs to be estimated from the classifier C (see Section 3.1.3 for details), and its label for learning the generator is re-formulated by including a real/fake discrimination node as below:

$$\tilde{y}^{ts} = [\overbrace{\underbrace{0, 0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{c-i}}^{y^t}, 0], x^{ts} \in X^{ts}, cat(x^{ts}) = i, \tag{9}$$

where y^t with c nodes stands for the one-hot categorial coding of x^t, which is estimated by C.

The last node in $\tilde{y}^{st}$ and $\tilde{y}^{ts}$ are for the fake sample. As the images x^{st} and x^{ts} are expected to be real when optimizing the generator G, the last node is set as 0 in Equations (8) and (9).

3.1.2 Duplex Discriminators

The function of the duplex discriminators is to distinguish the real images from the fake images, and also to categorize the real source and target images. In the overall scheme

of the proposed DupGAN, the duplex discriminators are stacked on the generator G to ensure the images generated from the generator look real and their category information are preserved (e.g., an image of "1" from one domain is still "1" but not one of the other categories when transformed to the other domain), which can further enforce the latent representation $E(x)$ domain invariant and informative. As it is difficult to directly constrain the label consistence of unlabeled target domain images, we first provide a pseudo label for the real target domain image x^t (detailed in Section 3.1.3), and then force its generated sample x^{ts} to be with the same label.

Specifically, the discriminator D^s for source domain attempts to distinguish the samples with source domain style, i.e., the real image x^s and the generated image x^{ts}. Besides, D^s also categorizes of the real image x^s. Thus, the output of D^s is a softmax layer with $c+1$ nodes, where the first c nodes represent the category for the real images, and the last one indicates the falsity of the input image. Similar with E and G, both D^s and D^t can any kind of deep neural network and their parameters are denoted as W^D.

When the input is x^s, D^s should classify it into one of the c real classes, i.e., the objective label is the known categorial label of images from X^s, detailed as follows:

$$\tilde{y}^s = [\overbrace{\underbrace{0,0,\cdots,0}_{i-1},1,\underbrace{0,\cdots,0}_{c-i}}^{y^s},0], x^s \in X^s, cat(x^s) = i. \tag{10}$$

When the input is x^{ts}, D^s should classify it into the fake class. The objective label is as follows:

$$\tilde{y}^{ts} = [\underbrace{0,0,\cdots,0}_{c},1], x^{ts} \in X^{ts}. \tag{11}$$

Like $\tilde{y}^{ts}$ in Equation (9), the labels $\tilde{y}^s$ and $\tilde{y}^{ts}$ are one-hot coding. When the input image is real, the node of its real class is set as 1 and others including the $(c+1)$-th node which indicates the falsity are set as 0. When the input image is fake, i.e., generated, the $(c+1)$-th node is set as 1 and others, i.e., the real class nodes, are set as 0.

Similarly, the discriminator D^t attempts to do real/fake discrimination and category classification for target domain. The input for D^t consists of images from X^t and X^{st}, and the output is also modeled as a softmax layer with $c+1$ nodes. The objective labels for the input images x^t and x^{st} are as follows:

$$\tilde{y}^t = [\overbrace{\underbrace{0,0,\cdots,0}_{i-1},1,\underbrace{0,\cdots,0}_{c-i}}^{y^t},0], x^t \in X^t, cat(x^t) = i, \tag{12}$$

$$\tilde{y}^{st} = [\underbrace{0,0,\cdots,0}_{C},1], x^{st} \in X^{st}. \tag{13}$$

The categorial labels of source domain images are known, however, that of target domain images are unavailable. Therefore, here the category information of target domain images are estimated from the classifier C which is stacked on the latent representation z (see Section 3.1.3 for details).

In summary, the objective function of the duplex discriminators D^s and D^t is formulated as below:

$$\begin{aligned} L_D &= \min_{W^D}\Big(\sum\nolimits_{x^s \in X^s} \big(H(D^s(x^s),\tilde{y}^s) + H(D^t(x^{st}),\tilde{y}^{st})\big) \\ &\qquad + \sum\nolimits_{x^t \in X^t} \big(H(D^t(x^t),\tilde{y}^t) + H(D^s(x^{ts}),\tilde{y}^{ts})\big)\Big) \\ &= \min_{W^D}\Big(\sum\nolimits_{x^s \in X^s} \big(H(D^s(x^s),\tilde{y}^s) + H(D^t(G(E(x^s),t)),\tilde{y}^{st})\big) \\ &\qquad + \sum\nolimits_{x^t \in X^t} \big(H(D^t(x^t),\tilde{y}^t) + H(D^s(G(E(x^t),s)),\tilde{y}^{ts})\big)\Big), \end{aligned} \tag{14}$$

where $H(\cdot,\cdot)$ is the cross entropy loss, and W^D represents the parameters of D^s and D^t.

It should be noted that the $\tilde{y}^{st}$ and $\tilde{y}^{ts}$ are equipped with different values during adversarial optimization between the generator and duplex discriminators. When optimizing D^s and D^t, the activations of X^{st} and X^{ts} are expected to lie in the last node, as in Equations (11) and (13), to enforce both of them capable of distinguishing the real images from the fake ones. When optimizing the generator G, the activations of X^{st} and X^{ts} are expected to lie in the first c nodes, as in Equations (10) and (12), to enforce the generator capable of producing real images.

3.1.3 Classifier

For the categorial classification, a classifier C is established on the latent representation z and its objective function is as follows:

$$L_C = \min_{W^C}\Big(\sum\nolimits_{x^s \in X^s} H(z^s, y^s) + \sum\nolimits_{x^t \in X^t} H(z^t, y^t)\Big), \tag{15}$$

where C can be constructed with any kind of deep network layers with softmax output, $H(\cdot,\cdot)$ is the cross entropy loss, and W^C represents the parameters of C. The categorial label y^s of the source domain image, which has been illustrated in Equations (8) and (11), are known and the pseudo label y^t of target domain, which has been illustrated in Equations (9) and (12), is estimated via the classifier C, which are both with c nodes w.r.t. one-hot coding. It is notable that the second term in Equation (15) only contains those pseudo-labeled samples with high confidence. With the premise that the highly confident pseudo labels are mostly correct [6, 40] and the dominance of commonality [2], y^t can be also used to train C without performance degradation. The classifier C needs to be pre-trained with only source domain images to ensure the initial transfer capability, i.e., the ability to obtain mostly correctly pseudo-labeled target domain samples with high confidence, due to the dominance of commonality in domain adaptation. Thus, the number of

samples in the second term is quite few or even zero at the very beginning and becomes larger as the training goes on.

Moreover, C further helps to attain domain-invariant z, because the domain-specific part in z will be discarded or concealed by G to fool both D^s and D^t with not only the real/fake discrimination but also the category classification. Thus, only the domain-invariant part in z can be exploited by D^s and D^t, and z tends to be domain-invariant to avoid information loss in G.

3.1.4 Overall Objective

The overall objective function can be formulated as follows:

$$L = \min_{W^E, W^C, W^G, W^D} \big(L_G + L_D + \beta L_C\big), \tag{16}$$

where β is a balance parameter.

The generator and the duplex discriminators are optimized in adversarial manner, and this can ensure the images from generator are real and their category information preserved. Like all adversarial learning methods, the labels of generated images for optimizing G and D are different as shown in Equations (8), (9), (11) and (13). Therefore, the overall network is optimized in an alternative way, i.e., alternative gradient descent w.r.t. W^D and $\{W^E, W^G, W^C\}$. The detailed optimization process is shown in Algorithm 1.

Algorithm 1 Optimization procedure of DupGAN.

Input: The source domain sample x^s and its category label y^s, target domain sample x^t without category label.

Output: The parameters of whole network, $W = \{W^E, W^C, W^G, W^D\}$.

1: Pre-train E and C with images in X^s.
2: **while** not converge **do**
3: Provide pseudo label for those images $x^t \in X^t$ with highly confident label prediction via C;
4: Update W^D by minimizing L_D in Equation (14) through the gradient descent:
$$W^D \leftarrow W^D - \eta \frac{\partial L_D}{\partial W^D}$$
5: Update W^C, W^G and W^E by minimizing $L_G + \beta L_C$ (in Equations (7) and (15)):
$$W^C \leftarrow W^C - \eta\beta \frac{\partial L_C}{\partial W^C}$$
$$W^G \leftarrow W^G - \eta \frac{\partial L_G}{\partial W^G}$$
$$W^E \leftarrow W^E - \eta(\beta \frac{\partial L_C}{\partial W^C} \times \frac{\partial W^C}{\partial W^E} + \frac{\partial L_G}{\partial W^G} \times \frac{\partial W^G}{\partial W^E})$$
6: **end while**

3.2. Difference from the Related Work

Difference from DANN [12, 13] and ADDA [48]. Both DANN and ADDA map the target domain samples to the source domain in the deep feature space where the adversarial loss of domain classification is applied. Both of them can not promise that the structure of target domain feature space is not distorted when mapped to source domain. On the contrary, our DupGAN not only alleviates the domain discrepancy but also preserves the category structure of target domain via the duplex discriminators with additional classification task. Moreover, our DupGAN is capable of image transformation between domains, while DANN and ADDA are not.

Difference from DRCN [15]. DRCN combines the classification task of source domain and the reconstruction task of target domain to find the shared feature space of the two domains, however, the shared representation is more in preference of source domain. More favorably, our DupGAN employs the adversarial learning between the generator and duplex discriminators to explicitly ensure the latent representation domain invariant.

Difference from DTN [47], CoGAN [30] and UNIT [29]. All of DTN, CoGAN, UNIT and our DupGAN follow the idea of generative adversarial network to achieve the cross domain representation and domain transformation. However, all the other three methods only conduct adversarial learning of real/fake which may cause structure distortion in the process of domain transformation. Differently, in our DupGAN, the domain adversarial learning is coupled together with the category classification, leading to domain invariant latent representation and domain transformation with category information undistorted.

Difference from kNN-Ad [42] and ATDA [40]. Both kNN-Ad and ATDA achieve unsupervised domain adaptation via re-labeling the unlabeled target domain. In both methods, the labeling for target domain is mainly based on the source domain samples without explicitly considering the domain discrepancy. In our DupGAN, the labeling for target domain is based on the domain invariant latent representation which can achieve more confident category labels. Besides, our method can conduct domain transformation, while both kNN-Ad and ATDA can not.

4. Experiments

The evaluation is to do category classification on the testing set of target domain with training on labeled source domain samples and unlabeled target domain ones. The performance is reported in terms of rank-1 accuracy of the classifier C. We compare the proposed DupGAN with a few state-of-the-art methods, including DANN [12, 13], ADDA [48], DSN [4], DRCN [15], CoGAN [30], UNIT [29], kNN-Ad [42] and ATDA [40] on digit classification task (i.e., MNIST $\leftrightarrow$ USPS and SVHN $\leftrightarrow$ MNIST detailed in Section 4.2.1) and object recognition task (i.e., Office-31 detailed in Section 4.2.2).

For more intensive comparison, we also evaluate the DCNN trained with only labeled source domain images, denoted as "DCNN-SourceOnly", and the DCNN trained with only labeled target domain images, denoted as "DCNN-TargetOnly". Both "DCNN-SourceOnly" and "DCNN-TargetOnly" are constructed with the same network structures as ours, and indicate the lower bound and upper bound performance of the domain adaptation.

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST	MNIST → SVHN	SVHN^{extra} → MNIST
DCNN-SourceOnly	86.75	75.52	62.19	33.7	73.67
DANN [12, 13]	85.1	73.0	73.85	-	-
DRCN [15]	91.8	73.67	81.97	40.05	-
ADDA [48]	92.87	93.75	76.0	-	86.37
DSN [4]	91.3	73.2	82.7	-	-
CoGAN [30]	95.65	93.15	-	-	-
UNIT [29]	95.97	93.58	-	-	90.53
kNN-Ad [42]	-	-	78.8	40.3	-
ATDA [40]	93.17	84.14	85.8	52.8	91.45
DupGAN(Ours)	**96.01**	**98.75**	**92.46**	**62.65**	**96.42**
DCNN-TargetOnly	95.02	98.96	98.97	87.74	98.97

Table 1: The results of unsupervised domain adaptation of digit classification. Because we share the same experiment settings with most compared works, we copy the corresponding results from the original papers. As for ADDA and DSN on MNIST ↔ USPS, SVHN^{extra} → MNIST and USPS → MNIST respectively, we directly use their released code to obtain the results as none are reported in the original works. For ATDA on MNIST ↔ USPS and SVHN^{extra} → MNIST, we implement it by ourselves to report the results as no code is released. The results signed with "-" are neither reported in original work nor tuned to reasonable performances. The SVHN^{extra} → MNIST experiment use the extra training set of SVHN.

4.1. Implementation Details

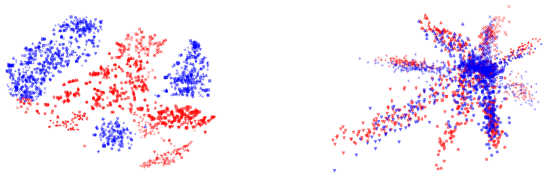
In all experiments, the pixel values in the input images are re-scaled to $[-1.0, 1.0]$. TanH function, i.e., $\frac{e^x - e^{-x}}{e^x + e^{-x}}$, is used as the activation function of the last layer in the generator G for scaling the output pixels to $[-1.0, 1.0]$, to be consistent with the input. The balance parameters α and β in MNIST ↔ USPS and SVHN → MNIST are empirically set as 10.0 and 0.2, respectively, and that in MNIST → SVHN are set as 1.0 and 1.0, respectively. In all the experiments of object recognition, α and β are set as 1.0 and 1.0, respectively. All the used architectures are the same with the state-of-the-art methods, detailed in the supplementary material.

The pseudo label y^t (mentioned in Section 3.1.3) with softmax score higher than a threshold is selected to train the classifier C. The threshold is set as 0.99 in MNIST ↔ USPS, SVHN → MNIST and all the experiments of object recognition, and that of MNIST → SVHN is set as 0.9 to avoid too few pseudo-labeled samples as this task is much harder.

4.2. Experiment Results

4.2.1 Unsupervised Domain Adaptation of Digit Classification

For digit classification, the datasets of MNIST [28], USPS [10] and SVHN [35] are used for evaluating a all the methods. All three datasets contain images of digits $0 \sim 9$ but with different styles. MNIST is composed of 60000 training and 10000 testing images. USPS consists of 7291 training and 2007 testing images. SVHN contains 73257 training, 26032 testing and 531131 extra training images. In the evaluation, we follow the same protocol with the compared methods for fair comparison. Specifically, in the experiments of MNIST ↔ USPS and SVHN → MNIST, we use similar network structure with that described in UNIT [29], and in the experiment of MNIST → SVHN, we use similar structure with [40].



(a) Before adaptation. (b) After adaptation.

Figure 3: The distribution of source and target domain samples before and after the adaption on SVHN → MNIST. (a) shows the distribution of the original source and target domain samples; (b) shows the distribution of the latent representation of the source and target domain samples obtained by our proposed DupGAN. The source and target domain samples are shown in red and blue respectively, and the categories are drawn with different shapes.

The performance of all methods are shown in Table 1. As seen, our proposed DupGAN outperforms all the compared methods, especially on the challenging MNIST → SVHN with an improvement up to about 10%. This superiority benefits from the adversarial learning between the generator and the duplex discriminators, which can ensure the latent representation domain invariant and informative. To deserve to be mentioned, DupGAN outperforms the "DCNN-TargetOnly" model, which may be due to that the source domain has a lot of labeled data with more variations better for adaptation, while the target domain has only limited labeled data, which is exactly the use case for unsupervised domain adaptation.

We further visualize the distribution of the learnt latent representation to investigate the effect of domain adaptation

Figure 4: The exemplars of domain transformation from SVHN to MNIST. In every two columns, the left and right images are the images from real source domain and their corresponding transformed images in target domain.

in the SVHN → MNIST experiment. As seen from Figure 3, the discrepancy is significantly reduced in the latent representation space of our proposed DupGAN. Besides, the category structure of the samples from both domain are well preserved as expected, which is beneficial for the final classification. Moreover, Figure 4 visualizes domain transformed images of our method from SVHN to MNIST dataset. As seen, our method can effectively achieve image transformation with category information preserved.

Ablation Study. We also try ablation study on unsupervised domain adaptation on MNIST → USPS and SVHN → MNIST with different components ablation, i.e., training with only the classifier C and the reconstruction loss (denoted as DupGAN-woA), training completely without the discriminators (denoted as DupGAN-woAD), and training with only the classifier C (denoted as DupGAN-woADG). The results are shown in Table 2.

Experiments	DupGAN	DupGAN-woA	DupGAN-woAD	DupGAN-woADG
MNIST → USPS	96.01	94.57	93.82	93.32
SVHN → MNIST	92.46	68.30	67.43	60.18

Table 2: Ablation study on digit classification of MNIST → USPS.

As can be seen, when one or more parts are removed, the performance degrades. The more parts are removed, the worse the performance is. The result also shows that all the parts are designed reasonably and they are in harmony with each other, forming an effective solution for domain adaptation, especially for those challenging cases.

4.2.2 Unsupervised Domain Adaptation on Object Recognition

We also evaluate our DupGAN on Office-31 dataset [39]. Office-31 is a standard benchmark for domain adaptation, consisting of 4110 images within 31 categories collected from three distinct domains: Amazon (A), Webcam (W) and DSLR (D). In this experiment, we only evaluate on the challeging settings of A ↔ W and A ↔ D. We follow the standard unsupervised domain adaptation training protocol, i.e., using all labeled source data and unlabeled target data, and employ the same network architecture as DRCN [15] (detailed in the supplementary material). For reconstructing the input images is harder in this senario, the generator aims for generating the feature map, as in DRCN [15]. The evaluation results are shown in Table 3. As can be seen, DupGAN outperforms the compared state-of-the-art methods, which again demonstrates the effectiveness of our proposed DupGAN.

Method	A → W	W → A	A → D	D → A
DCNN	61.6±0.5	49.8±0.4	63.8±0.5	51.1±0.6
DAN [31]	68.5±0.4	53.1±0.3	67.0±0.4	54.0±0.4
DANN [12, 13]	72.6±0.3	52.7±0.2	67.1±0.3	54.5±0.4
DRCN [15]	68.7±0.3	54.9±0.5	66.8±0.5	56.0±0.5
DupGAN(Ours)	**73.2**±0.2	**59.1**±0.5	**74.1**±0.6	**61.5**±0.5

Table 3: The results of unsupervised domain adaptation on Office-31. For the same experiment settings, we directly copy the results of the related work from the original papers. The "DCNN" are the results of AlexNet only finetuned with labeled source domain samples, copied from DAN [31].

5. Conclusion and Further Work

We propose a generative adversarial network with duplex discriminators named DupGAN to handle the unsupervised domain adaptation problem. DupGAN consists of an encoder, a generator and duplex discriminators, which respectively aim for encoding the input images to the latent representation, decoding the latent representation to source and target domains, and doing the category classification and real/fake discrimination. A classifier stacked on the encoder is used to classify samples from both the two domains directly. Benefited from the adversarial learning between the generator and the duplex discriminators, the latent representation is encouraged to domain invariant and category informative. As a result, the generator can achieve favorable domain transformation with less category distortion. As evaluated on several classical datasets, our proposed DupGAN achieves the state-of-the-art performance on unsupervised domain adaptation of digit classification and object recognition. In the future, we will explore the DupGAN on larger datasets with fine grained classification.

Acknowledgement

This work was partially supported by Natural Science Foundation of China under contracts Nos. 61390511, 61772496, and 61402443.

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 769–776, 2013.

[2] S. Ben-David et al. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[3] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *IEEE Journal of Machine Learning Research (JMLR)*, 10(9):2137–2155, 2009.

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 343–351, 2016.

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[6] M. Chen, K. Q. Weinberger, and J. C. Blitzer. Co-training for domain adaptation. In *International Conference on Neural Information Processing Systems*, pages 2456–2464, 2011.

[7] G. E. Dahl and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio Speech and Language Processing (ASLP)*, 20(1):30–42, 2012.

[8] W. Dai, G. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 210–219. ACM, 2007.

[9] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the IEEE International Conference on Machine learning (ICML)*, pages 193–200, 2007.

[10] J. S. Denker, W. R. Gardner, H. P. Graf, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, H. S. Baird, and I. Guyon. Neural network recognizer for hand-written zip code digits. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 323–331, 1988.

[11] M. Dudık, R. E. Schapire, and S. J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 323–330, 2005.

[12] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, and *et al*. Domain-adversarial training of neural networks. *IEEE Journal of Machine Learning Research (JMLR)*, 17(59):1–35, 2016.

[14] B. Geng, D. Tao, and C. Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing (TIP)*, 20(10):2980–2989, 2011.

[15] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 597–613. Springer, 2016.

[16] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[17] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Machine learning (ICML)*, pages 222–230, 2013.

[18] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[20] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006. IEEE, 2011.

[21] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the IEEE International Conference on Machine learning (ICML)*, pages 1764–1772, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[24] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, B. Schölkopf, and *et al*. Correcting sample selection bias by unlabeled data. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, volume 19, page 601. MIT; 1998, 2007.

[25] H. D. III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[27] A. Kumar, A. Saha, and H. Daume. Co-regularization based semi-supervised domain adaptation. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 478–486, 2010.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

[30] M. Liu and O. Tuzel. Coupled generative adversarial networks. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 469–477, 2016.

[31] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the IEEE International Conference on Machine learning (ICML)*, pages 97–105, 2015.

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 136–144, 2016.

[33] T. Luan, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, pages 1283–1292, 2017.

[34] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems Workshop (NIPSW)*, volume 2011, page 5, 2011.

[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks (TNN)*, 22(2):199–210, 2011.

[37] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (KDE)*, 22(10):1345–1359, 2010.

[38] G. Perarnau, V. D. W. Joost, B. Raducanu, and J. M. lvarez. Invertible conditional gans for image editing. 2016.

[39] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 213–226, 2010.

[40] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *arXiv preprint arXiv:1702.08400*, 2017.

[41] G. Saon, T. Sercu, S. Rennie, and H. K. J. Kuo. The ibm 2016 english conversational telephone speech recognition system. 2016.

[42] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2110–2118, 2016.

[43] M. Shao, C. Castillo, Z. Gu, and Y. Fu. Low-rank transfer subspace learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1104–1109. IEEE, 2012.

[44] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *IEEE International Journal of Computer Vision (IJCV)*, 109(1-2):74–93, 2014.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] M. Sugiyama, M. Krauledat, and K.-B. MÃžller. Covariate shift adaptation by importance weighted cross validation. *IEEE Journal of Machine Learning Research (JMLR)*, 8(5):985–1005, 2007.

[47] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[48] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

[49] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. The microsoft 2016 conversational speech recognition system. 2016.

[50] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the IEEE International Conference on Machine learning (ICML)*, page 114. ACM, 2004.

[51] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, pages 2242–2251, 2017.