Partially Adversarial Learning and Adaptation

Jen-Tzung Chien
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

Yu-Ying Lyu
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

***Abstract*—An image classification system for a specific target domain is usually trained with initialization from a source domain given with a large number of classes, particularly in an application of image recognition. The classes in target domain are usually seen as a subset in source domain. *Partial* domain adaptation aims to tackle this generalization issue where no labeled data are provided in target domain. This paper presents an adversarial learning for partial domain adaptation where a symmetric metric based on the Wasserstein distance is adopted in an adversarial learning objective. We build a Wasserstein partial transfer network where the Wasserstein adversarial objective is jointly optimized to partially transfer the relevance knowledge from source to target domains. The geometric property for optimal transport is assured to mitigate the gradient vanishing problem in adversarial training. The neural network components for feature extraction, relevance transfer, domain matching and task classification are jointly trained by solving a minimax optimization over multiple objectives. Experiments on image classification show the merits of the proposed *partially adversarial domain adaptation* over different tasks.**

***Index Terms*—image classification, domain adaptation, deep learning, adversarial learning, partial transfer**

I. Introduction

Deep learning has achieved a great success in many real-world applications ranging from computer vision to natural language processing where a large amount of labelled data are required for supervised training. Transfer learning provides a solution to utilize the knowledge of source domain to improve the learning performance for target domain [1], [2]. The costs for labelling data and retraining model can be significantly reduced [3]. Further, domain adaptation is known as a practical realization of transfer learning where the training data and class labels are provided in source domain but only test data are available in target domain. The goal of domain adaptation aims to perform data transforming and distribution matching between two domains so as to minimize the domain shift existing in raw data. Most of domain adaptation methods assume that both source and target domains have identical classes and only focus on learning the whole dataset distribution under this assumption. In [4], an useful approach was proposed for knowledge transfer from one *large* domain to another *small* domain. The overlapped classes in both domains were selected for domain adaptation where *negative* transfer was avoided.

In addition, the adversarial learning has incorporated to improve domain adaptation for image recognition, semantic segmentation, style transfer, etc. In [5], the domain adversarial network (DAN) was proposed by adding an adversarial objective in training procedure for domain adaptation. The measure of disparity between the distributions in two domains was minimized through a deep discriminator model. Adversarial learning plays a similar role in previous method for domain adaptation based on the maximum mean discrepancy (MMD) [6]–[8]. The architecture of DAN consisted of a feature encoder, a domain discriminator and a task classifier which were jointly optimized to extract domain invariant features for image recognition. The adversarial learning was performed so that the generated features were not clearly recognized between source and target domains. In [9], the Wasserstein distance guided representation learning was proposed. Different from DAN, the metric in domain discriminator was replaced by the Wasserstein distance which assures the preservation of gradient values for parameter updating when the distributions of features in source domain and target domain differed considerably in their manifolds. The domain discriminator could still provide sufficient gradient value for backpropagation for updating the whole model. This method was seen as a Wasserstein variant of DAN, also denoted as the WDAN. Different from DAN [5] and WDAN [9], this study presents a new Wasserstein adversarial network which deals with the issue of non-identical classes in image domain adaptation. The identical part and the redundant part in both domains are characterized by a relevance network without limitation on the number of classes in source domain. Partial transfer is performed to improve image recognition.

II. Background Survey

A. Adversarial Learning

Generative adversarial network (GAN) [10] is recognized as a powerful generative model where an implicit distribution is estimated for data generation. GAN consists of a generator G and a discriminator D that compete mutually in a two-player game based on a minimax adversarial optimization procedure. The generator samples a latent variable $\mathbf{z}$ from a standard Gaussian $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and use this sample to generate a synthesized sample $\widehat{\mathbf{x}}$. The discriminator is used to distinguish whether the input sample is from real data with $p_{\text{data}}(\mathbf{x})$ or synthetic data with $p_{\text{gen}}(\widehat{\mathbf{x}})$ using prior $p(\mathbf{z})$. The minimax optimization is formulated. The goal of GAN is to train a generator G that can map latent code $\mathbf{z}$ to a synthesized sample $\widehat{\mathbf{x}} \in \mathbb{R}^D$, and shape the generated data distribution $p_{\text{gen}}(\widehat{\mathbf{x}})$ to be close to real data distribution $p_{\text{data}}(\mathbf{x})$. The Jensen-Shannon divergence between two distributions was demonstrated as the

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

learning objective in vanilla GAN [10]. Such an *asymmetric* divergence causes the difficulty in training procedure due to gradient vanishing and mode collapse. In [11], f-GAN was developed by using the variational estimation where JS divergence in vanilla GAN is generalized to the f-divergence between two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$

$$\mathcal{D}_f(p||q) = \int q(\mathbf{x}) \sup_{t \in \text{dom}_{f^*}} \left\{ t\frac{p(\mathbf{x})}{q(\mathbf{x})} - f^*(t) \right\} d\mathbf{x}. \quad (1)$$

Different GANs are realized by designing different class of mapping functions $\mathcal{T}$ or choosing different f with a convex conjugate f^* from the domain dom_{f^*}. However, in the implementation, samples of these two distributions should be drawn to infer $\frac{p(\mathbf{x})}{q(\mathbf{x})}$ for f-GAN. If two distributions have no overlapped region, $\mathcal{D}_f(p||q)$ becomes intractable for GAN which results in an unstable optimization process where Nash equilibrium in minimax optimization was hard to achieve [12].

B. Adversarial Domain Adaptation

Adversarial domain adaptation was first introduced in [5] where the latent codes of source domain $\mathbf{z}_s$ and target domain $\mathbf{z}_t$ were jointly learned to compensate the domain shift according to a minimax optimization using their original data samples $\mathbf{x}_s$ and $\mathbf{x}_t$, respectively. The architecture of domain adversarial network (DAN) include a feature encoder or extractor E, a task classifier C_{task} and a domain discriminator D_{dom}. DAN aims to impose the encoder to generate domain invariant features. The adversarial learning is run to estimate the encoded features $\{\mathbf{z}_s, \mathbf{z}_t\}$ which are difficult to tell whether those features are from source or target domain. The learning objective using n_s training samples and class outputs $\{\mathbf{x}_{sn}, \mathbf{y}_{sn}\}$ in source domain, and n_t training samples $\{\mathbf{x}_{tn}\}$ in target domain is formulated as a minimax optimization over two losses

$$\min_{E, C_{\text{task}}} \max_{D_{\text{dom}}} \underbrace{-\sum_{n=1}^{n_s} \mathbf{y}_{sn} \log(C_{\text{task}}(E(\mathbf{x}_{sn})))}_{\mathcal{L}_c(E, C_{\text{task}})} + \underbrace{\sum_{n=1}^{n_s} \log D_{\text{dom}}(E(\mathbf{x}_{sn})) + \sum_{n=1}^{n_t} \log(1 - D_{\text{dom}}(E(\mathbf{x}_{tn})))}_{\mathcal{L}_d(E, D_{\text{dom}})}. \quad (2)$$

The first loss function is measured as the cross entropy error [13] for task classifier while the second loss function is calculated as the negative cross entropy error for binary classification over source and target domains. No labels are given in target domain. DAN aims to train a feature extractor E which produces the features for source domain $\mathbf{z}_s$ and target domain $\mathbf{z}_t$, and simultaneously train a task classifier C_{task} which achieves the lowest cross entropy error for the classes of $\{\mathbf{x}_{sn}\}$, and train a domain discriminator D_{dom} which attains the lowest cross entropy error for the domains of $\{\mathbf{x}_{sn}, \mathbf{x}_{tn}\}$.

C. Wasserstein Domain Adversarial Network

In [14], [15], the issues of mode collapse and gradient vanishing in GAN were addressed by presenting the reliable solutions especially when the distributions of feature representation of both domains $p(\mathbf{z}_s)$ and $p(\mathbf{z}_t)$ differ significantly in the manifold [11]. The key idea is to find the optimal transport from an original distribution to a target distribution. The classification problem in the discriminator of GAN turns out to deal with a regression problem for optimal transport. The p^{th} Wasserstein distance between two probability measures u and v as a metric of *optimal transport* is defined as

$$W_p(u, v) = \left(\inf_{\pi \in \Omega(u,v)} \int \int d(x, y)^p d\pi(x, y) \right)^{\frac{1}{p}} \quad (3)$$

where Ω is a space of u and v. Wasserstein GAN (WGAN) [14] was proposed by using the Wasserstein distance between p and q via the Kantorovich-Rubinstein duality under the 1-Lipschitz condition

$$W(p, q) = \sup_{\|f\|_{\text{Lip}} \leq 1} \mathbb{E}_{\mathbf{x} \sim p}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim q}[f(\mathbf{x})]. \quad (4)$$

In addition, the sliced Wasserstein (SW) distance [16] was proposed to project high dimensional probabilities into the sets of one-dimensional distributions, and then compare their one-dimensional representation by Wasserstein distance. The concept of the p^{th} sliced Wasserstein distance is to obtain the marginal distribution for high-dimensional (or d-dimensional) probability distribution p_x or p_y through linear projection

$$SW_p(p_x, p_y) = \left(\int_{\mathbb{S}^{d-1}} W_c(\mathcal{R}p_x(\cdot, \theta), \mathcal{R}p_y(\cdot, \theta)) d\theta \right)^{\frac{1}{p}} \quad (5)$$

where $\theta \in \mathbb{S}^{d-1}$, $\mathbb{S}^{d-1}$ is an unit sphere in $\mathbb{R}^d$, c is a transportation cost and $\mathcal{R}(\cdot, \theta)$ denotes the Radon transform which is an integral transform with the linearity property.

In [9], the Wasserstein distance guided representation learning was presented by incorporating the Wasserstein metric into domain adaptation. Similar to DAN [5], this method implemented a shared feature extractor to generate the domain invariant features. Key idea was to adopt the Wasserstein distance as the learning metric to learn the feature extractor as well as the domain discriminator. Thus, even when $p(\mathbf{z}_s)$ differed from $p(\mathbf{z}_t)$, there was still sufficient gradient calculated in backpropagation procedure for updating the parameters of domain discriminator. The weaknesses of gradient vanishing and mode collapse in GAN construction was mitigated. Without loss of generality, the resulting solution is herein called the Wasserstein domain adversarial network (WDAN).

III. Partially Adversarial Domain Adaptation

This study presents a Wasserstein adversarial network for image domain adaptation where the *partial learning* is developed to deal with the challenge when the number of classes in target domain is smaller than that in source domain. The overall system architecture is depicted in Figure 1. Source domain data $\{\mathbf{x}_s\} \in \mathbb{R}^{d \times n_s}$ are drawn from source distribution $p_s(\mathbf{x})$ while target domain data $\{\mathbf{x}_t\} \in \mathbb{R}^{d \times n_t}$ are drawn from target distribution $p_t(\mathbf{x})$. This system consists of source encoder E_s, target encoder E_t, classification network C_{task}, adaptation network D_{dom} and relevance network R. Different

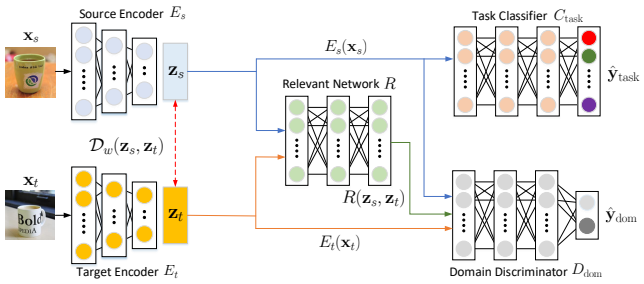

Fig. 1: Partially adversarial domain adaptation consists of source encoder E_s, target encoder E_t, relevance network R, task classifier C_{task} and domain discriminator D_{dom}.

from DAN and WDAN, the relevance network is merged and implemented for partial domain adaptation. This network measures the relevance of those source samples to target domain for partial transfer. In addition, two individual autoencoders E_s and E_t are estimated to extract individual latent codes $\mathbf{z}_s$ and $\mathbf{z}_t$ from source sample $\mathbf{x}_s$ and target sample $\mathbf{x}_t$, respectively. The reconstruction error is further minimized. The representation of latent codes is strengthened to carry out the following three networks for partially adversarial domain adaptation, which is seen as a partial transfer variant of WDAN (also simply denoted as the PWDAN). In general, the existence of generalization bound for partial domain adaptation using Wasserstein distance can be illustrated by referring to [15].

A. Classification Network

First of all, the classification network C_{task} is constructed as a task classifier based on the autoencoders E_s and E_t for source and target domains, respectively. Previous methods [4], [5], [9], [17] used only one shared feature encoder to learn joint representation. In order to cope with the non-identical domain data, we use the individual feature extractor and measure the Wasserstein distance between the encoded features $\mathbf{z}_s$ and $\mathbf{z}_t$ in both domains. Such a distance $W(\mathbf{z}_s, \mathbf{z}_t)$ is minimized to encourage the encoders to extract *domain invariant* features. The learning objective is constructed as a cross entropy error function $\mathcal{L}_c(E_s, C_{\text{task}})$ between the true label $\mathbf{y}$ and the predicted class $\hat{\mathbf{y}}_{\text{task}}$ from C_{task} using the features $E_s(\mathbf{x})$ from source distribution $p_s(\mathbf{x}, \mathbf{y})$

$$\min_{E_s, C_{\text{task}}} \underbrace{-\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_s(\mathbf{x},\mathbf{y})} \left[\mathbf{y} \log C_{\text{task}}(E_s(\mathbf{x}))\right]}_{\mathcal{L}_c(E_s, C_{\text{task}})}. \tag{6}$$

Target data have no labels and are excluded in (6). In addition, the feature extractors in both domains are estimated by minimizing the encoder loss $\mathcal{L}_{\text{enc}}(E_s, E_t)$ to achieve the smallest reconstruction error $\mathcal{L}_{\text{rec}}$ due to autoencoders $\{E_s, E_t\}$ in $\mathbf{x}$ space as well as the smallest Wasserstein distance $\mathcal{D}_w$ in $\mathbf{z}$ space

$$\min_{E_s, E_t} \underbrace{\mathcal{L}_{\text{rec}}(\mathbf{x}, \widehat{\mathbf{x}}) + \lambda_c \mathcal{D}_w(\mathbf{z}_s, \mathbf{z}_t)}_{\mathcal{L}_{\text{enc}}(E_s, E_t)} \tag{7}$$

where λ_c is a regularization parameter in classification network and $\widehat{\mathbf{x}}$ denote the reconstructed data for $\mathbf{x} = \{\mathbf{x}_s, \mathbf{x}_t\}$ where $\widehat{\mathbf{x}}_s = E_s(\mathbf{x}_s)$ and $\widehat{\mathbf{x}}_t = E_t(\mathbf{x}_t)$. Importantly, we adopt the sliced Wasserstein distance $\mathcal{D}_w \leftarrow SW_p$ and treat this distance as a regularization term for reconstruction to fulfill the Wasserstein variant of adversarial domain adaptation.

B. Relevance Network

To handle the challenge of partial transfer learning, it is important to design a *feature selection* mechanism to compensate for negative transfer. The relevance network is constructed to measure how relevant a sample $\mathbf{x}_s$ or a corresponding feature $\mathbf{z}_s$ in source domain exists in target domain. A discriminator network is trained to provide the probability of a sample $\mathbf{x}$ (either $\mathbf{x}_s$ or $\mathbf{x}_t$) existing in the *source* domain $p(y_{\text{dom}} = 1|\mathbf{x})$. The training is run by minimizing the loss of relevance network R or equivalently the discrimination error over the highly correlated samples or features. In traditional GAN, the output of optimal discriminator is $D^*(\mathbf{x}) \to 0$ if a sample is totally different from real data and $D^*(\mathbf{x}) \to 1$ if a sample is classified as real data. For partial domain adaptation, we focus on the feature selection over $\mathbf{z}_s$ and $\mathbf{z}_t$. $R^*(\mathbf{z}) \to 0$ means that latent code of source data $\mathbf{z}$ has high probability to be a shared class with target domain. In contrast, $R^*(\mathbf{z}) \to 1$ means that the relevance of a source feature $\mathbf{z}$ to a shared class with target domain is low. The relevance value is therefore calculated by

$$r(\mathbf{z}) = 1 - R^*(\mathbf{z}) = 1 - \frac{p_s(\mathbf{z})}{p_s(\mathbf{z}) + p_t(\mathbf{z})} = \frac{p_t(\mathbf{z})}{p_s(\mathbf{z}) + p_t(\mathbf{z})} \tag{8}$$

which has a value in a range of $[0, 1]$. Partial transfer is performed by estimating the relevance network R for distribution matching which minimizes the Wasserstein distance between the distributions of target and source domains

$$\begin{aligned} \min_{E_s, E_t, R} \; & \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}[1 - D_{\text{dom}}(E_t(\mathbf{x}))] \\ & -\mathbb{E}_{\mathbf{x}\sim p_s(\mathbf{x})}[r(\mathbf{z}) D_{\text{dom}}(E_s(\mathbf{x}))] \end{aligned} \tag{9}$$

where the Wasserstein distance in (4) is implemented. Notably, the relevance value $r(\mathbf{z})$ is introduced as a re-weighting factor for distribution measured from source data $\mathbf{x} \sim p_s(\mathbf{x})$. The Wasserstein distance in (9) is applied to learn the domain discriminator D_{dom} for adversarial domain adaptation.

C. Adaptation Network

The adaptation network is constructed for feature matching and domain adaptation which depend on the autoencoders E_s and E_t, the relevance network R and the domain discriminator D_{dom}. Inputs of domain discriminator are composed of the features from E_s and E_t, and the relevance value calculated by the relevance network R. The feature encoders and the domain discriminator $\{E_s, E_t, D_{\text{dom}}\}$ form an adversarial framework. D_{dom} learns a metric to minimize the divergence between $\mathbf{z}_s$ and $\mathbf{z}_t$ for feature matching. Feature extractors of two domains E_s and E_t are learned to produce the confusing features where the domain discriminator D_{dom} could not distinguish. The learning procedure is run to find the converged parameters for five networks $\{E_s, E_t, R, C_{\text{task}}, D_{\text{dom}}\}$ to achieve the goals of data reconstruction, partial transfer, pattern classification, feature matching and domain adaptation.

Algorithm 1 Training procedure for partially adversarial domain adaptation with Wasserstein metric

Initialize parameters $\Theta = \{\theta_s, \theta_t, \theta_r, \theta_c, \theta_d\}$
while θ_s, θ_t not converged **do**
  sample a minibatch $\{\mathbf{x}_s, \mathbf{y}_s, \mathbf{x}_t\}$ from a dataset
  apply autoencoders $\mathbf{z}_s \leftarrow E_s(\mathbf{x}_s)$, $\mathbf{z}_t \leftarrow E_t(\mathbf{x}_t)$
  calculate classifier loss $\mathcal{L}_c(E_s, C_{\text{task}})$
  update task classifier $\theta_c \leftarrow \theta_c - \eta \nabla_{\theta_c} \mathcal{L}_c$
  calculate reconstruction error $\mathcal{L}_{\text{rec}}(\mathbf{x}, \widehat{\mathbf{x}})$
  calculate regularization term $\mathcal{D}_w(\mathbf{z}_s, \mathbf{z}_t)$
  calculate encoder loss $\mathcal{L}_{\text{enc}}(E_s, E_t)$
  calculate discriminator loss $\mathcal{L}_d(E_s, E_t, R, D_{\text{dom}})$
  calculate gradient penalty $\mathcal{L}_{\text{grad}}(E_s, E_t)$
  update source autoencoder $\theta_s \leftarrow \theta_s$
    $-\eta \nabla_{\theta_s} \{\mathcal{L}_c + \mathcal{L}_{\text{enc}} + \mathcal{L}_d + \lambda_a \mathcal{L}_{\text{grad}}\}$
  update target autoencoder $\theta_t \leftarrow \theta_t$
    $-\eta \nabla_{\theta_t} \{\mathcal{L}_{\text{enc}} + \mathcal{L}_d + \lambda_a \mathcal{L}_{\text{grad}}\}$
  update domain discriminator $\theta_d \leftarrow \theta_d + \eta \nabla_{\theta_d} \mathcal{L}_d$
  update relevance network $\theta_r \leftarrow \theta_r - \eta \nabla_{\theta_r} \mathcal{L}_d$
end while

The loss function for domain discriminator is defined by the Wasserstein distance between source and target domains in latent spaces $\mathbf{z}_s$ and $\mathbf{z}_t$, respectively. Similar to the Wasserstein GAN in [18], we would like to meet the 1-Lipschitz constraint in Wasserstein distance so as to stabilize the training procedure for the proposed PWDAN. Such a constraint is relaxed by merging a gradient penalty $\mathcal{L}_{\text{grad}}(E_s, E_t)$ in a domain discriminator loss $\mathcal{L}_d(E_s, E_t, R, D_{\text{dom}})$ for a minimax optimization

$$\min_{E_s, E_t, R} \max_{D_{\text{dom}}} \Bigg\{ \underbrace{\mathbb{E}_{\mathbf{x} \sim p_t(\mathbf{x})}[1 - D_{\text{dom}}(E_t(\mathbf{x}))] - \mathbb{E}_{\mathbf{x} \sim p_s(\mathbf{x})}[r(\mathbf{z}) D_{\text{dom}}(E_s(\mathbf{x}))]}_{\mathcal{L}_d(E_s, E_t, R, D_{\text{dom}})} + \lambda_a \underbrace{\mathbb{E}_{\widehat{\mathbf{x}} \sim (p_s(\widehat{\mathbf{x}}) \cup p_t(\widehat{\mathbf{x}}))} (\|\nabla D_{\text{dom}}(\widehat{\mathbf{x}})\|_2 - 1)^2}_{\mathcal{L}_{\text{grad}}(E_s, E_t)} \Bigg\} \tag{10}$$

where λ_a is regularization parameter in adaptation network. Importantly, the discriminator posterior of source domain D_{dom} is multiplied by the relevance value $r(\mathbf{z})$ which is implemented for partial transfer. Such a partial transfer produces the optimal discriminator as $D^*_{\text{dom}} = \frac{r(\mathbf{z}) p_s(\mathbf{z})}{r(\mathbf{z}) p_s(\mathbf{z}) + p_t(\mathbf{z})}$ instead of $D^*_{\text{dom}} = \frac{p_s}{p_s + p_t}$ for full transfer using the standard GAN. Partial learning aims to learn the distributions of both domains to satisfy the matching condition $p_t(\mathbf{z}) \approx r(\mathbf{z}) p_s(\mathbf{z})$. Partial adaptation network is implemented in Algorithm 1.

IV. Experiments

The partially adversarial domain adaptation with Wasserstein objective (PWDAN) was implemented and compared with several domain adaptation methods including the deep neural network (DNN) with semi-supervised learning, the DAN [5], the DAN with Wasserstein metric (WDAN) [9], and the selective adversarial network (denoted as SAN) [4]. DNN was a baseline method without adversarial learning. The negative transfer happened in DAN and WDAN. SAN performed the positive transfer where the domain discriminator was separate for different classes in source domain and the weights of learning objective due to out-of-domain classes were decreased. For a fair comparison, we further implemented a new SAN with Wasserstein distance (also denoted as WSAN). DAN and SAN adopted the traditional GAN based on f-divergence. There are several novelties by using the proposed PWDAN. First, we introduced two separate autoencoders E_s and E_t and further minimized the reconstruction error $\mathcal{L}_{\text{rec}}$. Second, the sliced Wasserstein distance $\mathcal{D}_w(\mathbf{z}_s, \mathbf{z}_t)$ is minimized to match two domains in latent space. Third, a relevance network R is dedicated to find the relevance value $r(\mathbf{z})$ for re-weighting in partial domain adaptation. Four, the Wasserstein metric was used to estimate the model discriminator. Different from WSAN, PWDAN minimized the reconstruction error $\mathcal{L}_{\text{rec}}$ and regularization $\mathcal{D}_w(\mathbf{z}_s, \mathbf{z}_t)$, and used the relevance network R for partial transfer.

A. Experimental Setup

Three learning tasks were carried out for image recognition where the domain adaptation was performed by using five datasets. Three datasets (MNIST, MNIST-M and USPS) contained the images of handwritten digits under different distortions and environments with 10 classes while the remaining two datasets (Caltech and Office) had the images of objects collected from different sources and resolutions. Basically, the DNN or the encoders in E_s and E_t consisted of four convolutional and max-pooling layers with leaky ReLU activation given by slope 0.002 and kernel size 3x3, 5x5 or 7x7 depending on different datasets. λ_c and λ_a were empirically tuned. The relevance network R had three linear layers with ReLU activation and dropout. The discriminator contained three linear layers with ReLU activation. Batch normalization was applied in each layer in different networks. Three domain adaptation tasks MNIST $\rightarrow$ MNIST-M (M $\rightarrow$ MM), MNIST $\rightarrow$ USPS (M $\rightarrow$ U) and Caltech $\rightarrow$ Office (C $\rightarrow$ O) were examined. Different methods were investigated by full transfer as well as partial transfer. Full transfer was run by using the whole datasets. In evaluation of partial transfer using digit dataset, we removed the digits 0, 3, 6 and 9 and used the remaining 6 classes in target domain. In addition, the original Caltech contained 256 classes. For partial transfer, we selected those 10 categories in Office dataset from 31 classes, which were overlapped with those in Caltech dataset, The AlexNet [19] was used in the initialization.

B. Experimental Result

Figure 2 displays the distributions of 2-D visualization of original and partially transferred samples (MNIST $\rightarrow$ MNIST-M) embedded by t-SNE [20]. It is obvious that domain adaptation using DAN and PWDAN produces the matching samples in both domains. Both domains are separate in original data space. Further, PWDAN performs better than DAN because six classes in PWDAN are well separated compared with those in DAN. Table I lists the classification accuracy of target data using different methods under three tasks. It is found that the performance drops significantly in the setting

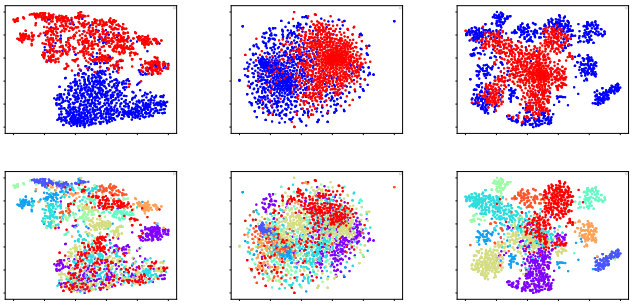

Fig. 2: Original distributions of MNIST (10 classes) (blue) and MNIST-M (6 classes) (red) (1st column). Distributions after partial transfer using DAN (2nd column) and PWDAN (3rd column). 1st and 2nd rows show the results for domains and classes, respectively. Six classes are indicated by colors.

	M → MM	M → U	C → O
DNN (F)	76.7	68.1	64.1
DAN (F)	86.5	75.4	73.6
WDAN (F)	88.5	80.2	75.3
SAN (F)	89.1	81.9	75.7
WSAN (F)	91.8	**84.8**	76.9
PWDAN (F)	**92.6**	84.1	**79.3**
DAN (P)	79.5	70.8	69.3
WDAN (P)	82.8	75.7	70.1
SAN (P)	85.6	79.8	74.0
WSAN (P)	87.9	82.3	75.2
PWDAN (P)	**89.0**	**83.9**	**77.1**

TABLE I: Comparison of classification accuracy (%) on target domain data under full transfer (F) and partial transfer (P).

of partial transfer. No matter the setting is full transfer or partial transfer, the proposed PWDAN consistently achieves higher accuracy than DAN and WDAN. SAN and WSAN performs better than DAN and WDAN, respectively. Domain discriminators for individual classes in SAN work well. The Wasserstein metric used in WDAN and WSAN receives the increased accuracy compared with the f-divergence in DAN and WDAN. The proposed PWDAN outperforms SAN and WSAN in most cases. The treatments of partial transfer using relevance network and gradient calculation based on Wasserstein distance work well. Such a result is consistently observed in three tasks of domain adaptation.

V. Conclusions

This paper has presented a new adversarial domain adaptation where the optimal transport was implemented in a partially adversarial learning procedure. The domain shift between source and target domains was minimized in accordance with the Wasserstein metric. The representative features in both domains were calculated with minimum reconstruction error to fulfill a stable calculation of gradient in error back-propagation. A relevance network was introduced to measure how likely a source sample appearing in target domain and use the relevance value to re-weight the learning objective. The partial transfer was carried out for the non-identical class issue in image classification. Multi-objective learning over different goals with regularization was performed. Three tasks of image domain adaptation were investigated. Experimental results assured the increase of classification accuracy due to the relevance network and the use of Wasserstein distances as the regularization term and learning criterion.

References

[1] H.-Y. Chen and J.-T. Chien, "Deep semi-supervised learning for domain adaptation," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

[2] J.-C. Tsai and J.-T. Chien, "Adversarial domain separation and adaptation," in *Prof. of IEEE International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[3] X. Cui, J. Huang, and J.-T. Chien, "Multi-view and multi-objective semi-supervised learning for hmm-based automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 1923–1935, 2012.

[4] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[6] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems 19*, pp. 513–520. MIT Press, 2007.

[7] W. Lin, M.-W. Mak, and J.-T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[8] W. Lin, M.-W. Mak, L. Li, and J.-T. Chien, "Reducing domain mismatch by maximum mean discrepancy autoencoders," in *Proc. of Speaker and Language Recognition Workshop*, 2018, pp. 162–167.

[9] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. of AAAI Conference on Artificial Intelligence*, 2018.

[10] 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*, 2014, pp. 2672–2680.

[11] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[12] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of GANs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1825–1835.

[13] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 361–374, 2016.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017.

[15] L. Redko, A. Habrard, and M. Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 737–753.

[16] S. Kolouri, G. K. Rohde, and H. Hoffmann, "Sliced Wasserstein distance for learning Gaussian mixture models," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.

[17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.

[18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105. 2012.

[20] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.