# Enhanced Transport Distance for Unsupervised Domain Adaptation

Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, Chuan-Xian Ren*
School of Mathematics, Sun Yat-Sen University, China
limx36@mail3.sysu.edu.cn, {zhaiym3, luoyw28, gepengf}@mail2.sysu.edu.cn, rchuanx@mail.sysu.edu.cn

## Abstract

*Unsupervised domain adaptation (UDA) is a representative problem in transfer learning, which aims to improve the classification performance on an unlabeled target domain by exploiting discriminant information from a labeled source domain. The optimal transport model has been used for UDA in the perspective of distribution matching. However, the transport distance cannot reflect the discriminant information from either domain knowledge or category prior. In this work, we propose an enhanced transport distance (ETD) for UDA. This method builds an attention-aware transport distance, which can be viewed as the prediction-feedback of the iteratively learned classifier, to measure the domain discrepancy. Further, the Kantorovich potential variable is re-parameterized by deep neural networks to learn the distribution in the latent space. The entropy-based regularization is developed to explore the intrinsic structure of the target domain. The proposed method is optimized alternately in an end-to-end manner. Extensive experiments are conducted on four benchmark datasets to demonstrate the SOTA performance of ETD.*

## 1. Introduction

Sufficient labeled data play an important role in training discriminative and interpretable models which have wide applications in pattern recognition and machine learning [24, 28]. However, not all tasks have plenty of labeled data for model training, and the collection of annotated data is very expensive and time-consuming. Fortunately, due to the explosive growth of information, the big data era provides us sufficient training data from multiple sources and scenarios, by which we can extract discriminative features and exploratory data structure. However, there exist large distribution gaps and domain shifts between different domains due to many factors (e.g., postures, locations and views) [5, 12]. Applying the model trained on the existing labeled source domain to the unlabeled domain directly will result in a significant performance degradation.

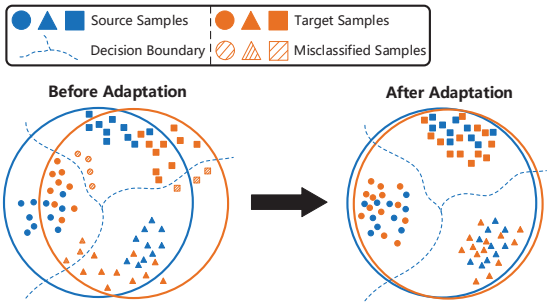



Figure 1. The primary goal of UDA is to generalize the well-trained classifier on the source domain to the unlabeled target domain. Direct application of the learned classifier suffered from the "domain shift" problem, as shown in the left case. UDA methods match the features of different domains, and then reduce the misclassification rate, as shown in the right case.

In unsupervised domain adaptation (UDA), the target domain data are unlabeled. As shown in Figure 1, UDA methods solve tasks in the target domain by exploiting the information from the labeled source domain. Those methods attempt to establish the association between different domains, and then learn a shared and domain-invariant feature space [7, 17, 30].

Optimal transport (OT) distance [32, 33], known as the Earth Mover's distance, is a classic metric that plays a fundamental role in the probability simplex. Compared with Kullback-Leibler or Jensen-Shannon divergence [8], OT distance is the unique one that needs to be parameterized. Though the support sets of two distributions do not overlap with each other, the OT distance can still measure the relation between the distributions [1].

It is worth noting that the OT model [32] has received wide attention in domain adaptation. It aims to learn the transformation between domains with theoretical guarantees [5, 23]. The optimal transport distance is used to learn an optimal transport plan, which plays an important role in mapping the data from the source domain to the target domain [29], so that the classifier trained on the source domain generalizes well in the target domain. Recently, Bhushan et

*Corresponding Author.

al. [3] employ deep neural networks to explore the optimal transport plan and further learn the domain-similar features.

Existing domain adaptation methods using the OT distance are mainly constrained by two bottlenecks. First, deep adaptation methods using the transport distance are based on the Euclidean metric and the mini-batch training manner. Since the sampled instances within mini-batches cannot fully reflect the real distribution, the obtained transport distance lacks discrimination and the estimated transport plan is biased. Second, some OT methods ignore the label information and latent structure of the target domain.

In order to tackle the above bottlenecks, this paper proposes a model called Enhanced Transport Distance (ETD) for classification problems in UDA. The basic model structure consists of three parts, i.e., the feature extraction network, the classification network and the optimization of transport plan. In particular, our novelty is emphasized in the third part. The main contributions of this paper are summarized as follows.

- To make full use of the category-based outputs of the classification network, we exploit the attention mechanism to estimate the similarity between samples, and weigh the transport distances with the attention scores. The weighed distances are expected to learn the discriminant features well.

- By virtue of the powerful fitting ability of deep networks, it is expected to learn the optimal transport plan with higher accuracy. The Kantorovich potential is re-parameterized by a three-layer fully connected network, instead of the vectored dual variable in the traditional semi-dual optimization problems.

- The entropy criterion on the target domain is used to explore the intrinsic structures of the target distribution. Thus, it improves the transferability of the learned classifier. The whole model is trained in an end-to-end manner. Extensive experiments conducted on benchmark datasets show that ETD achieves competitive performance in UDA tasks.

## 2. Related Work

The UDA models assume that the source domain has sufficient labels, and the unlabeled target domain participates the model training in an unsupervised manner [2]. The UDA methods can be divided into two categories roughly, i.e., methods based on shared features and methods based on data reconstruction.

*Methods based on shared features.* Early methods minimize distribution gaps by extracting shared features from different domains [12]. They often learn a feature projection to align different domains. Weight-sharing techniques are used to learn domain-invariant features through different metrics, e.g., Maximum Mean Discrepancy (MMD) [17, 20, 25]. Contrastive Adaptation Network (CAN) [13] proposes to optimize the intra-class and the inter-class domain discrepancies by a new metric. Deep Adaption Network (DAN) [17] adapts the high-layer features with the multi-kernel MMD criterion. Adversarial Discriminative Domain Adaptation (ADDA) [30] is trained in a minimax paradigm such that the feature extractor is learned to fool the classifier, while the classifier struggles to be discriminative. General to Adapt (GTA) [28] induces a symbiotic relation between the embedding network and the generative adversarial network. Conditional Domain Adversarial Network (CDAN) [18] is a principled framework that conditions the adversarial adaptation models on discriminative information conveyed in the classifier predictions.

*Methods based on reconstruction.* Reconstruction-based adaptation methods aim to learn a shared-latent representation between two domains and maintain the domain-specific characterizations [34]. These methods learn an encoder to extract the domain-invariant features, and maintain the representation consistency through minimizing the reconstruction error between domains. Domain Adversarial Neural Network (DANN) [7] proposes a domain-adversarial training to promote the emergence of features that are discriminative for the main learning task on the source domain while indiscriminate w.r.t. the domain shifts. Image to image translation for domain adaptation (I2I Adapt) [22] requires that the features extracted are able to reconstruct the images in both domains while the distribution of features in the two domains are indistinguishable. Unsupervised Image-to-Image Translation (UNIT) [16] makes a shared-latent space assumption and proposes an unsupervised image-to-image translation framework based on Coupled GANs.

The OT distance measures the distribution divergence and offers the UDA task a new viewpoint. However, these advantages come at a huge computational cost. Fortunately, Geneval et al. [9] propose a class of stochastic optimization schemes for obtaining the optimal transport distance for both discrete and continuous distributions.

## 3. Brief Review of Optimal Transport

Some important notations are defined here. Let $\mathcal{X}$ and $\mathcal{Z}$ be complete metric spaces, e.g., Euclidean space. We denote random multi-variates such as $X$ by capital letters, and use $X \sim \mu$ to indicate that $X$ obeys the probability measure $\mu$. $\boldsymbol{x}$ is a sample vector sampled from $\mu$. $\mathrm{Supp}(\mu)$ refers to the support of $\mu$, which is a subset of $\mathcal{X}$.

The OT distance is originally formulated as the Monge problem [32]. Considering two random variables $X \sim \mu$ and $Z \sim \nu$, which are sampled from space $\mathcal{X}$ and $\mathcal{Z}$, re-

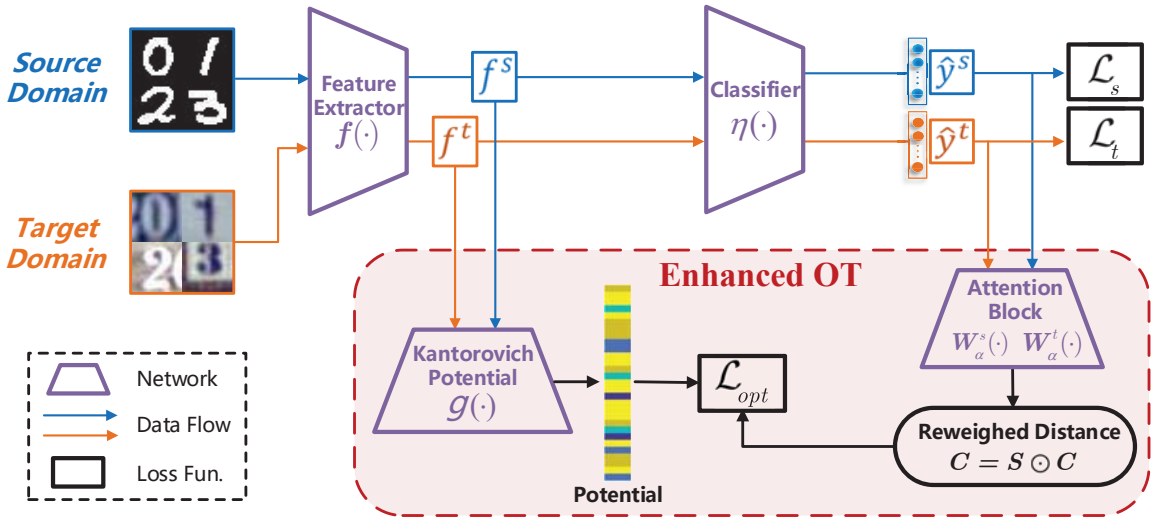



Figure 2. Adaptive model structure diagram for the proposed ETD. The source and target domains share the network weights of the feature extractor. The Kantorovich potential network is constructed by three fully-connected layers, and the attention network is formulated as a single fully connected layer. Blue and Orange arrows represent data flow of the source and target domain respectively. The attention matrix is used to reweigh the optimal transport distance.

spectively, and a loss function

$$c : (X, Z) \in \mathcal{X} \times \mathcal{Z} \mapsto c(X, Z) \in \mathbb{R}^+.$$

The goal of the OT problem is to find a mapping $\kappa : \mathcal{X} \mapsto \mathcal{Z}$ that maps the unit quality from $\mu$ to $\nu$ when the transport cost is minimized, i.e.,

$$\inf_{\kappa} \mathbb{E}_{X\sim\mu}[c(X, \kappa(X))] \text{ s.t. } \kappa(X) \sim Z. \tag{1}$$

To make the problem shown in Eq. (1) more feasible, Kantorovich [14] relaxed the Monge problem by casting it into a minimization over couplings $(X, Z) \sim \pi$ rather than the set of maps, where $\pi$ should have marginals equal to $\mu$ and $\nu$. The Kantorovich relaxation allows the mass at a given point $X \in \text{Supp}(\mu)$ to be transferred to several positions in $Z \in \text{Supp}(\nu)$.

The hard constraint for $\pi$ can be further relaxed, and the computation of OT can be speeded up by adding a strictly convex regularization term $R(\cdot)$, i.e.,

$$\inf_{\pi} \mathbb{E}_{(X,Z)\sim\pi}\left[c(X, Z)\right] + \varepsilon R(\pi) \text{ s.t. } X \sim \mu, Z \sim \nu, \tag{2}$$
$$R_\varepsilon(\pi) \triangleq \text{KL}(\pi|\mu \otimes \nu).$$

The regularization term also accelerates the optimization by making the OT distance differentiable everywhere w.r.t. the weights of the input measure [4].

## 4. Our Method

Assume that we can access to a source domain $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)|i = 1, 2, \cdots, n_s\}$ and a target domain $\mathcal{D}^t = \{\boldsymbol{x}_j^t|j = 1, 2, \cdots, n_t\}$. Here, $\boldsymbol{x}_i$ represents a sample (feature vector), and $y_i$ represents the corresponding label. Notice that the source domain data are labeled and the target domain data are unlabeled. Let $\hat{y}_j^t$ be the softmax prediction of the classifier, which can be updated in the training procedure. Since the model update is based on mini-batch, we suppose there is a training batch $\mathcal{B} = \mathcal{B}^s \bigcup \mathcal{B}^t$, which contains a source batch $\mathcal{B}^s = \{(\boldsymbol{x}_i^s, y_i^s)|i = 1, 2, \cdots, b\}$ and a target batch $\mathcal{B}^t = \{(\boldsymbol{x}_j^t, \hat{y}_j^t)|j = 1, 2, \cdots, b\}$. Here $b$ is the mini-batch size.

The entire network structure consists of three parts, namely, a feature extractor network $f(\cdot)$, a classification network $\eta(\cdot)$ and the Kantorovich potential network $g(\cdot)$. Their working flows are shown in Figure 2. The feature extractor takes original image $\boldsymbol{x}$ as input and outputs deep feature $f(\boldsymbol{x})$. The classifier maps the feature $f(\boldsymbol{x})$ to classification prediction $\eta(f(\boldsymbol{x}))$. The parametrization network returns the optimal transport plan $g(f(\boldsymbol{x}))$ based on the deep feature. Besides, we introduce an attention module to reweigh the transport distance.

### 4.1. Attention-based Distance Weighing

The distance function $c$ involved in the OT problem is used to calculate the distance between two samples via their features. Let $\boldsymbol{C}$ be the distance matrix for each batch[1], i.e.,

$$\boldsymbol{C}_{ij} = c(f_i^s, f_j^t).$$

Generally, the distance measurement is specified to the Euclidean metric due to its simplicity.

[1]From now on, the features of the source and target domains are abbreviated as $f^s$ and $f^t$, respectively.

As the OT problem is optimized within the mini-batch, the learned transport plan are inconsistent at each training iteration. We propose an adaptively weighed OT algorithm based on the attention mechanism. The attention score indicates the degree of correlation between samples. By applying the reweighed distance matrix, current mini-batch is adjusted to the real data distribution.

The attention network consists of two fully connected network layers. It takes $\eta(f^s)$ and $\eta(f^t)$ as inputs and outputs the attention matrix $\boldsymbol{S} \in \mathbb{R}^{b\times b}$, which will be used as the weighing operator. $\boldsymbol{S}_{ij}$ represents the correlation of the source domain sample $\boldsymbol{x}_i^s$ and the target domain sample $\boldsymbol{x}_j^t$. Let $\sigma(\cdot)$ be the activation function, such as sigmoid, tanh and ReLU. The attention matrix is formulated by

$$\boldsymbol{S}_{ij} = \sigma\left[(\boldsymbol{W}_a^s \eta(f_i^s)\,^T (\boldsymbol{W}_a^t \eta(f_i^t))\right], \tag{3}$$

where $\boldsymbol{W}_a^s$ and $\boldsymbol{W}_a^t$ are projection matrix of the one layer attention network. ReLU is used in our method due to its effectiveness in tackling the gradient vanishing and explosion problems. Moreover, ReLU is unilateral and has a constant derivation in most cases, which is helpful for faster convergence. Then the attention matrix $\boldsymbol{S}$ is normalized to unit length as $\sum_{j=1}^{b_t} \boldsymbol{S}_{ij} = 1$, e.g., by softmax, where $b_t$ is the mini-batch size of the target domain.

The update procedure of this attention network is included in the parameter update of the overall model. It should be noted that the calculation of the attention score is based on the current mini-batch. Thus, the weighing operator is also dynamically approximate to the complete data space. The reweighed distance matrix is formulated as

$$\boldsymbol{C} = \boldsymbol{S} \odot \boldsymbol{C}. \tag{4}$$

Here $\boldsymbol{C}$ is still used to denote the reweighed distance matrix without symbol abuse. It deduces a reweighed distance metric $c(f_i^s, f_j^t) = \boldsymbol{S}_{ij} c(f_i^s, f_j^t)$, which is used to redefine the optimal transport problem.

The adaptively adjusted distance function results in a transport plan that is expected to approximate the actual scenario. As for the optimal transform model, though the Sinkhorn algorithm [6] can solve the problem of Eq. (2), it does not scale well to metrics supported by a large number of samples. In order to tackle this problem, Fenchel-Rockafellar et al. [29] define

$$F_\varepsilon(u, v) \triangleq -\varepsilon \exp\left(\frac{u(X) + v(Z) - c(X, Z)}{\varepsilon}\right), \forall \varepsilon > 0.$$

Then the dual problem is derived from the dual theorem [29] as

$$\sup_{u,v} \mathbb{E}_{(X,Z)\sim(\mu\times\nu)}[u(X) + v(Z) + F_\varepsilon(u(X), v(Z))], \tag{5}$$

where $u$ and $v$ are known as Kantorovich potentials [4]. This dual approximation of the regularized OT problem can be effectively solved by Stochastic Gradient Descent (SGD) methods [1, 29].

The relation between $u$ and $v$ is obtained by writing the first order optimality condition for $v$. Then the relational expression and the reweighed distance matrix are inserted into Eq. (5) yields the semi-dual formulation of the reweighed OT problem [9], i.e.,

$$\sup_v \mathbb{E}_{f^s\sim\mu}[v^{c,\varepsilon}(f^s)] + \mathbb{E}_{f^t\sim\nu}[v(f^t)] - \varepsilon. \tag{6}$$

Our exploration of the reweighed OT problem and its domain adaptation extension is based on the above efforts.

### 4.2. Network Re-parametrization of the Kantorovich Potential

Instead of direct optimization of the transport plan, existing methods use the semi-dual variable $v$ as the optimization target, known as the Kantorovich potential [14]. They initialize the dual variable $v$ by a random vector (dimension equals the distributed sample size) and update it in an iterative manner [9]. In this paper, deep networks are used to parameterize the dual variable due to its strong fitting ability, rather than the original vector approaches.

The Kantorovich potential network $g$ consists of three fully connected layers, and it transforms features of two domains into the dual variable. Since update of the potential variable re-parametrized network $g$ is an independent loop outside the overall model optimization, the corresponding parameters are denoted as $\boldsymbol{W}_g$, and the parameters of other parts as $\boldsymbol{W}$.

The optimization problem w.r.t. the parameters $\boldsymbol{W}_g$ becomes

$$\sup_v \mathbb{E}_{f^s\sim\mu}[g^{c,\varepsilon}(f^s)] + \mathbb{E}_{f^t\sim\nu}[g(f^t)] - \varepsilon,$$

where

$$g^{c,\varepsilon}(f^s) = \begin{cases} -\varepsilon \log\left(\mathbb{E}_{f^t\sim\nu}\left[\exp\left(\frac{g(f^t) - c(f^s, f^t)}{\varepsilon}\right)\right]\right) & , \varepsilon > 0, \\ \min\limits_{f^t\in\mathcal{Z}} (c(f^s, f^t) - g(f^t)) & , \varepsilon = 0. \end{cases}$$

The network re-parametrization of the Kantorovich potential variable provides a more powerful fitting and generalization ability, and the optimization is efficient. The applicability of dual variable is extended by such an approach, so that the optimal transport plan learning procedure can get rid of the constraints of the current feature space and procees in a more flexible space.

To present the calculation clearly, the adaptively weighed OT problem is re-formulated as a finite dimensional optimization problem w.r.t the distance function $c(\cdot, \cdot)$, i.e.,

$$\max_g \mathcal{H}_\varepsilon(g(f^t)) = \sum_{i=1}^{b} \Big( \sum_{j=1}^{b} g(f_j^t) + h_\varepsilon(f_i^s, g(f^t)) \Big),$$

where

$$h_\varepsilon(f_i^s, g(f^t)) = \begin{cases} \varepsilon \log\left(\sum_{j=1}^{b} \exp\left(\frac{g(f_j^t) - c(f_i^s, f_j^t)}{\varepsilon}\right)\right) - \varepsilon & , \varepsilon > 0, \\ \min_j (c(f_i^s, f_j^t) - g(f_j^t)) & , \varepsilon = 0. \end{cases}$$

In this way, the optimization procedure of the transport plan $\pi^\varepsilon$ can be transformed into the optimization of the network $g$ in the training process. Optimizing the dual variable makes the training process more suitable for the mini-batch training scheme in deep networks. Besides, the network-based re-parametrization can simplify the algorithm calculation. After optimizing the network $g$, the optimal transport distance $W_\varepsilon(\mu, \nu)$ is calculated by the current parameters $\boldsymbol{W}_g$, i.e.,

$$W_\varepsilon(\mu, \nu) = \mathbb{E}_{f^s \sim \mu} g^{c,\varepsilon}(f^s) + \mathbb{E}_{f^t \sim \nu} g(f^t) - \varepsilon, \tag{7}$$
$$g^{c,\varepsilon}(f^s) = -\varepsilon \log\left(\mathbb{E}_{f^t \sim \nu}\left[\exp\left(\frac{g(f^t) - c(f^s, f^t)}{\varepsilon}\right)\right]\right).$$

Finally, the domain discrepancy (i.e., the alignment loss) is measured by the OT distance as

$$\mathcal{L}_{opt}(\boldsymbol{W}) \triangleq W_\varepsilon(\mu, \nu). \tag{8}$$

### 4.3. Discriminant Features Adaption and Model Optimization

According to the theoretical result of Ben-David [2], the expected error on the target domain is bound by three terms, i.e., the expected domain discrepancy, the expected error on the domain classifier, and a shared error (which is usually viewed as constant) of the idea joint hypothesis. Thus, we also need to minimize the classification error and extract discriminant features on the source domain. This part is essentially a supervised learning task only performed on the source domain. The cross entropy loss denoted by $l_{ce}$ is used here, due to its simplicity, i.e.,

$$\mathcal{L}_s(\boldsymbol{W}) \triangleq \frac{1}{n_s} \sum_{i=1}^{n_s} l_{ce}(\eta(f_i^s), y_i^s; \boldsymbol{W}),$$

where $\boldsymbol{W}$ represents all the parameters (excluding $\boldsymbol{W}_g$) updated in the adaptation network learning.

To explore the the intrinsic structure of the target domain, the entropy criterion is added to the objective function. Mathematically, the target entropy loss denoted by $\mathcal{L}_t$ is formulated as

$$\mathcal{L}_t(\boldsymbol{W}) \triangleq \frac{1}{n_t} \sum_{i}^{n_t} \sum_{j=1} -\hat{y}_{ij}^t \log \hat{y}_{ij}^t,$$

where $\hat{y}_{ij}^t$ is the probability of the $i$-th target samples belonging to the the $j$-th class.

Combining the optimization objective functions described above, the whole cost function of the model consists three parts, namely the source classification loss $\mathcal{L}_s(\boldsymbol{W})$, the domain adaptation loss $\mathcal{L}_{opt}(\boldsymbol{W})$ and the target entropy loss $\mathcal{L}_t(\boldsymbol{W})$, which can be written as

$$\mathcal{L}(\boldsymbol{W}) = \mathcal{L}_s(\boldsymbol{W}) + \lambda \mathcal{L}_{opt}(\boldsymbol{W}) + \beta \mathcal{L}_t(\boldsymbol{W}). \tag{9}$$

The parameter $\lambda$ and $\beta$ are used to balance the effects of three loss terms. The model adapts different domains by minimizing the optimal transport loss $\mathcal{L}_{opt}(\boldsymbol{W})$, and it learns a discriminant classifier by minimizing the cross entropy loss of the source domain $\mathcal{L}_s(\boldsymbol{W})$. Further, the target entropy loss $\mathcal{L}_t(\boldsymbol{W})$ helps the model to learn an adaptive classifier. The parameters will be learned by minimizing $\mathcal{L}$ with the SGD approach.

**Algorithm 1** ETD for Unsupervised Domain Adaptation

**Require:** $\mathcal{B}^s = \{\boldsymbol{x}^s; y^s\}$, $\mathcal{B}^t = \{\boldsymbol{x}^t\}$, $\boldsymbol{C} \in \mathbb{R}^{b \times b}$, $\varepsilon$, $\lambda$.
**Ensure:** $\boldsymbol{W}_g$, $\boldsymbol{W}$, $W_\varepsilon$, $\hat{Y}^t$.

```
Pre-train the network W and W_g by using B^s;
Predict the pseudo-labels {ŷ^t} for samples in B^t;
while not converged do
    Calculate attention S via Eq. (3) and softmax;
    Re-weigh C via Eq. (4), i.e., C = S ⊙ C;
    while not converged do
        Update W_g by minimizing W_ε via Eq. (7) and Eq. (8);
    end while
    Update W by minimizing Eq.(9);
end while
```

The main steps of our ETD method are summarized in Algorithm 1. Note that there are two loops in the algorithm, and they aim to optimize the parameters $\boldsymbol{W}$ and $\boldsymbol{W}_g$, respectively. The optimal transport module (w.r.t. $\boldsymbol{W}_g$) is a build-in loop in the network update process. we update parameters of the adaptive model and the Kantorovich potential re-parametrization in an alternative manner. First, we fix the network parameters $\boldsymbol{W}$ and determine the optimal transport plan, and then fix the transport plan $\boldsymbol{W}_g$ to update the network parameters. The convergent condition for each of them can be defined by the relative error between two successive iterations. In our work, the convergent condition is simply set by pre-defined iteration times. Note that the algorithm is illustrated based on each mini-batch, but in actual experiments, we can choose to make the network forward through one or more batches and then carry out the backpropagation of the whole network.

## 5. Experiment Results and Analysis

In this section, the proposed ETD is compared with several SOTA methods on four UDA benchmark datasets.

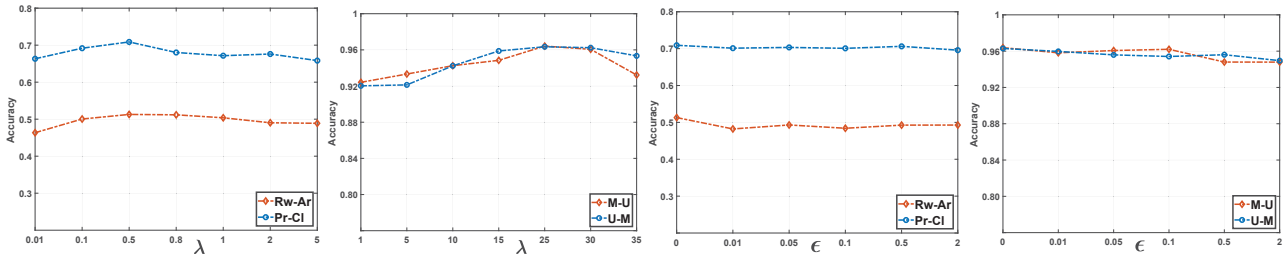



Figure 3. Classification results under different $\lambda$ and $\varepsilon$ settings. Best viewed in color.

### 5.1. Datasets, Settings and Implementation Details

The algorithm is evaluated by conducting experiments on four visual datasets.

**Office-31** [27] consists of images from three domains, namely, Amazon (A), Webcam (W), and Dslr (D). It has 4,652 images covering 31 categories. Six adaptation tasks, i.e., A→W, D→W, W→D, A→D, D→A and W→A, will be conducted in the experiments.

**ImageCLEF-DA** has twelve classes shared by three datasets, i.e., Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). All six adaptation tasks, i.e., I→P, P→I, I→C, C→I, C→P, P→C, will be used to algorithm evaluation.

**Office-Home** [31] consists of 15,500 images in 65 categories, mostly from an office or home environment. The images are sampled from four distinct areas including Art (Ar), Clipart (Cl), Product (Pr), and Real World (Rw). We evaluate the methods on all twelve learning tasks.

**Digits Recognition** This work uses an evaluation scheme being consistent with Cyclic Anti-Adaptive Network (CyCADA) [11], which includes three transfer tasks, i.e., SVHN to MNIST (S→M), USPS to MNIST (U→M) and MNIST to USPS (M→U). In particular, tasks U→M and M→U will be evaluated under the protocols of [7].

The network backbones and basic settings are described as follows. For the U→M and M→U tasks, LeNet [15] is used as the network backbone due to the small-sample-size of the training setting. Note that the images are of gray-scale, and the pixels are of 16×16 (USPS) and 28×28 (MNIST), respectively. The small-scaled images are interpolated to fit the large-sized images, and the single-channel images are replicated to fit the three-channel setting. For the remaining tasks, ResNet-50 [10] is used as the network backbone. The mini-batch size is 32. Empirically, the parameter $\beta$ of the target entropy loss in Eq. (9) is set as 1e-1 to reduce the risk of the uncertain predictions. The basic experimental setups follow a standard set of unsupervised domain adaptive settings [17].

In the experiments, LeNet is initialized by random values, and ResNet-50 is initialized by pre-training on the ImageNet [26]. We use Adam optimizer to train all layers of the network through back-propagation. The dimension of the bottleneck layer is set to 256.

Table 3. Accuracy (%) on *MNIST, USPS and SVHN* datasets (based on ResNet-50)

| Method | M→U | U→M | S→M | Avg |
|---|---|---|---|---|
| DANN [7] | 95.7±0.1 | 90.0±0.2 | 70.8±0.2 | 85.5 |
| UNIT [16] | **96.9**±0.3 | 93.6±0.2 | 90.5±0.2 | 93.4 |
| ADDA [30] | 89.4±0.2 | 90.1±0.8 | 76.0±1.8 | 85.2 |
| CyCADA [11] | 95.6±0.4 | **96.5**±0.2 | 90.4±0.3 | 94.2 |
| I2I Adapt [22] | 95.1±0.1 | 92.2±0.2 | 92.1±0.2 | 93.1 |
| GTA [28] | 95.3±0.3 | 90.8±0.2 | 92.4±0.1 | 92.8 |
| **ETD** | 96.4±0.3 | 96.3±0.1 | **97.9**±0.4 | **96.9** |

### 5.2. Parameter Sensitivity and Ablation Study

The ETD method has two important hyper-parameters, i.e., $\lambda$ and $\varepsilon$. The parameter $\lambda$ acts on the total loss function $\mathcal{L}$ to balance the classification loss $\mathcal{L}_s$ and the transport distance based domain discrepancy $\mathcal{L}_{opt}$. The parameter $\varepsilon$ is the regularization coefficient in the optimal transport problem. They correspond to the outer-loop and the inner-loop, respectively. Both of them are assumed to be non-negative.

The parameter sensitivity of ETD is evaluated on two datasets, i.e., *Office-Home* and *Digit Recognition*. We use the try-and-error approach by selecting parameter values from pre-defined sets. *Office-Home* have 65 classes while *Digit Recognition* have 10 classes. The former dataset has a more serious domain shift problem for UDA. Therefore, the optimal value of parameter $\lambda$ has different ranges for different tasks as illustrated in Figure 3. Parameter $\lambda$ is selected from $\{0.01, 0.1, 0.5, 0.8, 1, 2, 5\}$ for *Office-Home* and $\{1, 5, 10, 15, 25, 30, 35\}$ for *Digit Recognition*. The optimal parameter value for the *Office-Home* dataset is $\lambda = 0.5$, and for the *Digit Recognition* dataset is $\lambda = 25$. Parameter $\varepsilon$ is selected from the collection $\{0, 0.01, 0.05, 0.1, 0.5, 2\}$ for both *Office-Home* and *Digit Recognition*. We can see that the optimal parameter value on both datasets is $\varepsilon = 0$. Experimental results present that the proposal is stable under different parameter settings.

The ablation study is conducted on the *ImageCLEF-DA* dataset. The attention network, target entropy and Kan-

Table 1. Accuracy (%) on *Office-31* and *ImageCLEF-DA* datasets (based on ResNet-50)

| Method | *Office-31* | | | | | | | *ImageCLEF-DA* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A→W | D→W | W→D | A→D | D→A | W→A | Avg | I→P | P→I | I→C | C→I | C→P | P→C | Avg |
| ResNet[10] | 68.4 | 96.7 | 99.3 | 68.9 | 62.5 | 60.7 | 76.1 | 74.8 | 83.9 | 91.5 | 78.0 | 65.5 | 91.27 | 80.7 |
| DAN[17] | 80.5 | 97.1 | 99.6 | 78.6 | 63.6 | 62.8 | 80.4 | 74.8 | 83.9 | 91.5 | 78.0 | 65.5 | 91.27 | 80.7 |
| RTN[19] | 70.2 | 96.6 | 95.5 | 66.3 | 54.9 | 53.1 | 72.8 | - | - | - | - | - | - | - |
| DANN[7] | 84.5 | 96.8 | 99.4 | 77.5 | 66.2 | 64.8 | 81.6 | 75.0 | 86.0 | 96.2 | 87.0 | 74.3 | 91.5 | 85.0 |
| JAN[20] | 82.0 | 96.9 | 99.1 | 79.7 | 68.2 | 67.4 | 82.2 | 76.8 | 88.4 | 94.8 | 89.5 | 74.2 | 91.7 | 85.8 |
| GTA[28] | 89.5 | 97.9 | 99.8 | 87.7 | 72.8 | 71.4 | 86.5 | - | - | - | - | - | - | - |
| CDAN[18] | 93.1 | 98.2 | 100.0 | 89.8 | 70.1 | 68.0 | 86.6 | 76.7 | 90.6 | 97.0 | 90.5 | 74.5 | 93.5 | 87.1 |
| CDAN+E[18] | **94.1** | 98.6 | 100.0 | **92.9** | 71.0 | **69.3** | **87.7** | 77.7 | 90.7 | 97.7 | 91.3 | 74.2 | 94.3 | 87.7 |
| **ETD** | 92.1 | **100.0** | **100.0** | 88.0 | **71.0** | 67.8 | 86.2 | **81.0** | **91.7** | **97.9** | **93.3** | **79.5** | **95.0** | **89.7** |

Table 2. Accuracy (%) on *Office-Home* dataset (based on ResNet-50)

| Method | Ar→Cl | Ar→Pr | Ar→Rw | Cl→Ar | Cl→Pr | Cl→Rw | Pr→Ar | Pr→Cl | Pr→Rw | Rw→Ar | Rw→Cl | Rw→Pr | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet[10] | 34.9 | 50.0 | 58.0 | 37.4 | 41.9 | 46.2 | 38.5 | 31.2 | 60.4 | 53.9 | 41.2 | 59.9 | 46.1 |
| DAN[17] | 43.6 | 57.0 | 67.9 | 45.8 | 56.5 | 60.4 | 44.0 | 43.6 | 67.7 | 63.1 | 51.5 | 74.3 | 56.3 |
| DANN[7] | 45.6 | 59.3 | 70.1 | 47.0 | 58.5 | 60.9 | 46.1 | 43.7 | 68.5 | 63.2 | 51.8 | 76.8 | 57.6 |
| JAN[20] | 45.9 | 61.2 | 68.9 | 50.4 | 59.7 | 61.0 | 45.8 | 43.4 | 70.3 | 63.9 | 52.4 | 76.8 | 58.3 |
| CAN[13] | 50.8 | 71.5 | 74.8 | **58.0** | **70.6** | 70.0 | 57.3 | **52.5** | 75.9 | 69.3 | 56.2 | 80.9 | 65.7 |
| CDAN[18] | 49.0 | 69.3 | 74.5 | 54.4 | 66.0 | 68.4 | 55.6 | 48.3 | 75.9 | 68.4 | 55.4 | 80.5 | 63.8 |
| CDAN+E[18] | 50.7 | 70.6 | 76.0 | 57.6 | 70.0 | 70.0 | 57.4 | 50.9 | 77.3 | **70.9** | 56.7 | 81.6 | 65.8 |
| **ETD** | **51.3** | **71.9** | **85.7** | 57.6 | 69.2 | **73.7** | **57.8** | 51.2 | **79.3** | **70.2** | **57.5** | **82.1** | **67.3** |

torovich potential network are denoted by Att., $\mathcal{L}_t$ and KPN, respectively. To demonstrate the effectiveness of Kantorovich potential network $g$, we design a stacked version of OT. Specifically, this method takes the features of three fully connected layers as input and optimize the OT problem according to [9] independently.

### 5.3. Results and Comparative Analysis

The proposed method ETD is compared with the latest domain adaptive methods, e.g., DAN [17], Residual Transfer Network (RTN) [19], DANN [7], Joint Adaptation Network (JAN) [20], GTA [28], CAN [13], CDAN [18], UNIT [16], ADDA [30], CyCADA [11], and I2I Adapt [22].

Table 1 shows classification results on *Office-31* and *ImageCLEF-DA* datasets. Compared with CDAN [18], the accuracy of our method increases by 1.4% on task D→W, and the average accuracy of our model is 86.2% which is slightly lower but has basically reached the level of the most advanced methods. The average accuracy of our method is 89.7%. Compared with the most advanced method [18], the accuracy of our model increases by 3.3% on the task I→P, and the average accuracy of our model is 89.7% which is improved by 2%. The model average accuracy is superior to the latest methods in all tasks.

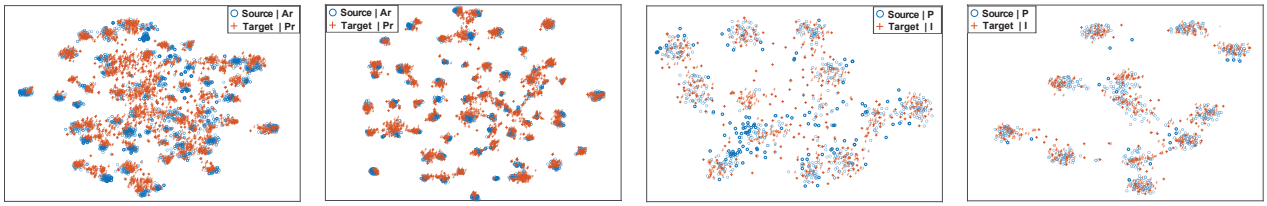
Table 2 shows classification results on *Office-Home* dataset. The average accuracy of our method is 67.3%, which exceeds the SOTA methods. Since there are more categories and larger domain discrepancy in *Office-Home*, model training is more difficult. Compared with CDAN [18], the accuracy of our model increases by 9.7% on task Ar→Rw, and the average accuracy increases by 1.5%.

Table 4. Accuracy (%) of ablation study on *ImageCLEF-DA*.

| stacked | Att. | $\mathcal{L}_t$ | KPN | I→C | I→P | P→C | P→I |
|---|---|---|---|---|---|---|---|
| ✓ | | | | 93.4 | 77.2 | 92.0 | 89.2 |
| ✓ | ✓ | | | 93.8 | 78.0 | 92.0 | 89.4 |
| ✓ | | ✓ | | <u>95.8</u> | <u>78.4</u> | 94.4 | 91.0 |
| ✓ | ✓ | ✓ | | 95.2 | 78.2 | **95.2** | **92.0** |
| | | ✓ | ✓ | 96.22 | 78.5 | 94.5 | 91.2 |
| | ✓ | ✓ | ✓ | **98.0** | **81.0** | <u>95.0</u> | <u>91.7</u> |

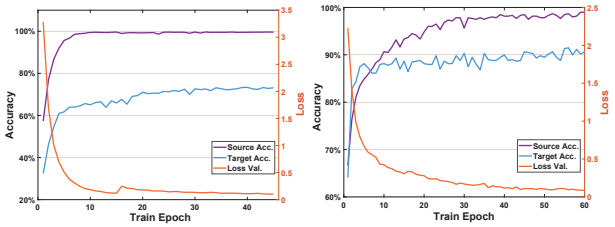
Table 3 shows classification results on *Digits Recognition* datasets. Experiments on this dataset consist of 3 tasks with two network structures. It can be observed that the proposed method improves the average accuracy by 2.7% and achieves the highest accuracy in the task S→M.

In Table 4, the accuracies of stacked with $\mathcal{L}_t$ on tasks P→I and P→C are improved about 1.0% by adding the attention module. As for other tasks, the improvement of the attention module is less significant. This is probably due to the existing consistency between OT distance and the results of the learned classifier. Compared with the stacked module with $\mathcal{L}_t$ increases 0.5%∼1.0% almost on all tasks. The accuracies of KPN with $\mathcal{L}_t$ are further improved by adding the attention module.




(a) Before adaptation (*Office-Home*) (b) After adaptation (*Office-Home*) (c) Before adaptation (*ImageCLEF*) (d) After adaptation (*ImageCLEF*)

Figure 4. Feature Visualization of *Office-Home* and *ImageCLEF-DA*. Best viewed in color.




(a) Art→Product (b) P→I

Figure 5. Classification accuracy and loss value w.r.t. different epochs. Best viewed in color.

### 5.4. Visualization and Training Stability

To present the adaptive learning process more intuitively, t-distribution Stochastic Neighbour Embedding (t-SNE) [21] is used to show the changes in low-dimensional visualization before and after the adaptation learning. We use two tasks, i.e., Ar→Pr (*Office-home*) and I→P (*ImageCLEF-DA*), to conduct the experiments. The results are shown in Figure 4, where (a)-(b) correspond to task Ar→Pr and (c)-(d) task P→I. Apparently, the spatial distribution of different domains is quite different before adaptive learning. It indicates that the distribution has no discriminative structure and is evenly dispersed in the feature space. However, the feature distribution shows an obvious cluster structure after domain adaptive learning. Cluster centers of distributions are closer than before, and the dispersion degree is more similar. It means that our method significantly changes the distribution in the feature space.

To show the training stability of the algorithm, we present the classification accuracy and training loss w.r.t. different epochs. We select two tasks, i.e., Ar→Pr (*Office-Home*) and P→I (*ImageCLEF-DA*) as instances to illustrate the performance of ETD, and present the results in Figure 5. These results are recorded in the stage of fine-tuning, instead of pre-training. The left diagram shows the experiment results of task Ar→Pr. We can see that the loss values drop very quickly and then begin to stabilize within the first ten epochs. It tends to be stable when more epochs are used for training. The classification accuracy fluctuates as the epoch size increases, but the overall trend is upward. The right diagram shows the results of task P→I. The loss drops rapidly at the beginning of the training process. It then drops relatively smooth when more epochs are used for training, and the overall trend is downward. The classification accuracy increases rapidly in the first ten epochs, and then the speed slows down.

## 6. Conclusion and Future Work

In this paper, we propose an end-to-end method called ETD to tackle the bottlenecks of OT in UDA. Specifically, ETD develops an attention-aware OT distance to measure the domain discrepancy under the guidance of the prediction-feedback. The Kantorovich potential is re-parameterized by deep neural networks to make the transport plan of OT more precise. Further, the domain discrepancy is minimized by reducing the transport costs. To make the most use of prediction information and explore the distribution structures, the entropy criterion is applied to the target domain. Experimental results illustrate the superiority of the ETD compared with other SOTA methods. How to apply the re-parameterized OT method to other practical problems, such as image generation, target detection, and object tracking, is our future work.

## 7. Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 61976229, 61906046, 11631015, and U1611265.

## References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, pages 137–144, 2007.

[3] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, pages 447–463, 2018.

[4] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 880–889, 2018.

[5] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[6] Charlie Frogner and Tomaso Poggio. Fast and flexible inference of joint distributions from their marginals. In *ICML*, pages 2002–2011, 2019.

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

[8] Pengfei Ge, Chuan Xian Ren, Dao Qing Dai, Jiashi Feng, and Shuicheng Yan. Dual adversarial autoencoders for clustering. *IEEE Transactions on Neural Networks and Learning System*, 2019, Accepted.

[9] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *NIPS*, pages 3440–3448, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

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

[12] I Hong Jhuo, Dong Liu, DT Lee, and Shih Fu Chang. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, pages 2168–2175, 2012.

[13] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[14] Leonid V Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382, 2006.

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

[16] Ming Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, pages 700–708, 2017.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[18] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NIPS*, pages 1647–1657, 2018.

[19] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pages 136–144, 2016.

[20] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[22] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, pages 4500–4509, 2018.

[23] Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In *NIPS*, pages 4197–4205, 2016.

[24] Chuan Xian Ren, Pengfei Ge, Dao Qing Dai, and Hong Yan. Learning kernel for conditional moment-matching discrepancy-based image classification. *IEEE Transactions on Cybernetics*, 2019, Accepted.

[25] Chuan Xian Ren, Xiao Lin Xu, and Hong Yan. Generalized conditional domain adaptation: A causal perspective with low-rank translators. *IEEE Transactions on Cybernetics*, pages 1–14, 2018, Accepted.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[27] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[28] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, 2018.

[29] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

[30] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[31] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[32] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[33] Zhen Zhang, Mianzhi Wang, and Arye Nehorai. Optimal transport in Reproducing Kernel Hilbert Spaces: Theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, Accepted.

[34] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251, 2017.