Selective Transfer with Reinforced Transfer Network for Partial Domain Adaptation

Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, Xinyu Jin
Institue of Information Science and Electronic Engineering, Zhejiang University
{zhihongchen,chench,chengzhaowei,byjiang,ke-fang,jinxy}@zju.edu.cn

Abstract

One crucial aspect of partial domain adaptation (PDA) is how to select the relevant source samples in the shared classes for knowledge transfer. Previous PDA methods tackle this problem by re-weighting the source samples based on their high-level information (deep features). However, since the domain shift between source and target domains, only using the deep features for sample selection is defective. We argue that it is more reasonable to additionally exploit the pixel-level information for PDA problem, as the appearance difference between outlier source classes and target classes is significantly large. In this paper, we propose a reinforced transfer network (RTNet), which utilizes both high-level and pixel-level information for PDA problem. Our RTNet is composed of a reinforced data selector (RDS) based on reinforcement learning (RL), which filters out the outlier source samples, and a domain adaptation model which minimizes the domain discrepancy in the shared label space. Specifically, in the RDS, we design a novel reward based on the reconstruct errors of selected source samples on the target generator, which introduces the pixel-level information to guide the learning of RDS. Besides, we develope a state containing high-level information, which used by the RDS for sample selection. The proposed RDS is a general module, which can be easily integrated into existing DA models to make them fit the PDA situation. Extensive experiments indicate that RTNet can achieve state-of-the-art performance for PDA tasks on several benchmark datasets.

1. Introduction

Deep neural networks have achieved impressive performance in a variety of applications. However, when applied to related but different domains, the generalization ability of the learned model may be severely degraded due to the harmful effects of the domain shift [3]. Re-collecting labeled data from the coming new domain is prohibitive because of the huge cost of data annotation. Domain adaptation (DA) techniques solve such a problem by transferring knowledge from a source domain with rich labeled data to a target domain where labels are scarce or unavailable. These DA methods learn domain-invariant features by moment matching [18, 25, 9] or adversarial training [26, 12].

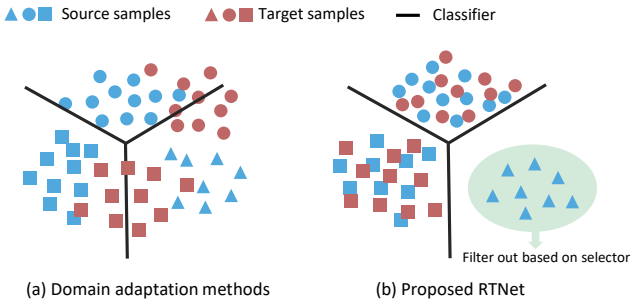


Figure 1: (a) Negative transfer is triggered by mismatch. (b) Negative transfer is mitigated by filtering out outlier classes.

Previous DA methods generally assume that the source and target domains have shared label space, i.e., the category set of the source domain is consistent with that of the target domain. However, in real applications, it is usually formidable to find a relevant source domain with identical label space as the target domain. Thus, a more realistic scenario is partial domain adaptation (PDA) [4], which relaxes the constraint that source and target domains share the same label space and assumes that the unknown target label space is a subset of the source label space. In such a scenario, as shown in Figure 1a, existing DA methods force an error match between the outlier source class (blue triangle) and the unrelated target class (red square) by aligning the whole source domain with the target domain. As a result, the negative transfer may be triggered due to the mismatch. Negative transfer is a dilemma that the transfer model performs even worse than the non-adaptation (NoA) model [21].

Several approaches have been proposed to solve the PDA

problem by re-weighting the source samples, where the weights can be get from the distribution of the predicted target label probabilities [5] or the prediction of the domain discriminator [4, 32]. These methods select the relevant source samples only considering the high-level information (deep features), which however ignore the most discriminative features hidden in the pixel-level information, such as appearance, style or background. Since the difference of appearance between the outlier source samples and the target samples is significantly large, taking into account the pixel-level information for outlier sample selection is expected to benefit the adaptation performance [13]. Moreover, these PDA modules based on adversarial networks are difficult to integrate into matching-based DA methods lacking discriminators. Therefore, most existing matching-based methods are hard to extend to address the PDA problem.

In this paper, to address the PDA problem, we present a reinforced transfer network (RTNet), as shown in Figure 1b, which exploits reinforcement learning (RL) to learn a reinforced data selector (RDS) for filtering outlier source samples. In this respect, the DA model from RTNet can align distributions in the shared label space to avoid negative transfer. To utilize both pixel-level and high-level information, we design a RDS. The RDS takes action (keep or drop a sample) based on the state of sample. Then, the reconstruction error of the selected source sample on the target generator is used as a reward to guide the learning of RDS via the actor-critic algorithm [15]. Note that, the state contains high-level information, and the reward contains pixel-level information. Specifically, the intuition of using reconstruction error to introduce pixel-level information is that the target generator lacks training samples of outlier classes and the outlier source samples extremely dissimilar to the target classes, so on the generator trained with target samples, the reconstruction error of outlier sample is larger than that of the related source samples. Hence, the reconstruction error can measure the appearance similarity between each source sample and the target domain well, which is the important information in sample selection and hard to get from high-level information.

The contributions of this work are: (1) a novel PDA framework RTNet is proposed, which joints sample selection and domain discrepancy minimization. (2) we design a reinforced data selector based on reinforcement learning, which solves the PDA problem by taking into account high-level and pixel-level information to select related samples for positive transfer. As far as we know, this is the first work to address PDA problem with RL technique. (3) most DA methods can be extended to solve PDA problem by integrating the RDS. We use two types of base network to evaluate the effectiveness of integration. (4) The RTNet achieves the best performance on three well-known benchmarks.

2. Related Work

Partial Domain Adaptation: Deep DA methods have been widely studied in recent years. These methods extend deep models by embedding adaptation layers for moment matching [27, 17, 25, 7, 8] or adding domain discriminators for adversarial training [12, 26]. However, these methods may be restricted by the assumption that source and target domains share the same label space, which is not held in PDA scenario. Several methods have been proposed to solve the PDA problem. Selective adversarial network (SAN) [4] trains a separate domain discriminator for each class with a weight mechanism to suppress the harmful influence of outlier classes. Partial adversarial domain adaptation (PADA) [5] improves SAN by adopting only one domain discriminator and gets the weight of each class based on the target probability distribution output by classifier. Example transfer network (ETN) [6] quantifies the weights of source examples based on their similarities to the target domain. Unlike previous PDA methods, only high-level information was used, RTNet combines pixel-level and high-level information to achieve more accurate sample filtering.

Reinforcement Learning: RL can be roughly divided into two categories [1]: value-based methods and policy-based methods. The value-based methods estimate future expected total rewards through a state, such as SARSA [23] and deep Q network [19]. Policy-based methods try to directly find the next best action in the current state, such as REINFORCE algorithm [30]. To reduce variance, some methods combine value-based and policy-based methods for more stable training, such as the actor-critic algorithm [15]. So far, data selection based on RL has been applied in the fields of active learning [11], co-training [31], text matching [22], etc. However, there is a lack of reinforced data selection methods to solve the PDA problem.

3. Our Approach

Problem Definition and Notations: In this work, based on PDA settings, we define the labeled source dataset as $\{\boldsymbol{X}^s, Y^s\} = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ from source domain $\mathcal{D}_s$ associated with $|\mathcal{C}_s|$ classes, and define the unlabeled target dataset as $\{\boldsymbol{X}^t\} = \{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$ from target domain $\mathcal{D}_t$ associated with $|\mathcal{C}_t|$ classes. Note that, the target label space is contained in the source label space, i.e., $\mathcal{C}_t \in \mathcal{C}_s$ and $\mathcal{C}_t$ is unknown. The two domains follow different marginal distributions, p and q, respectively, we further have $p_{\mathcal{C}_t} \neq q$. $p_{\mathcal{C}_t}$ is the distribution of source samples in the target label space. The goal is to improve the performance of model in $\mathcal{D}_t$ with the help of the knowledge in $\mathcal{D}_s$ associated with $\mathcal{C}_t$.

3.1. Overiew of RTNet

As shown in Figure 2, RTNet consists of two components: a domain adaptation model (F and C) and a rein-

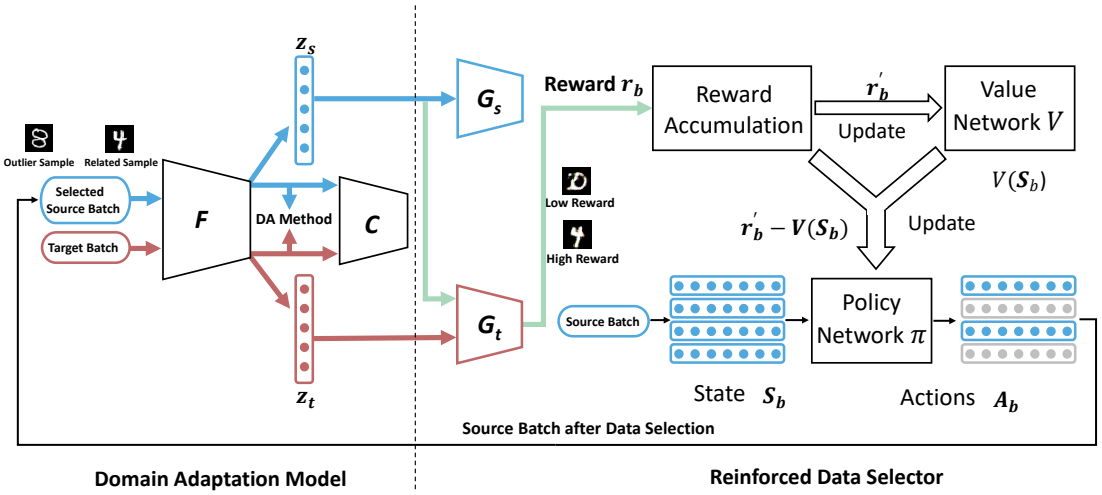


Figure 2: Overview of RTNet. F is a shared feature extractor, C is a shared classifier, G_s and G_t are source and target generators respectively, V is a value network and π is a policy network. G_s and G_t are combined with F to construct source and target auto-encoders to reconstruct samples, respectively. The green line indicates the flow to get the reward.

forced data selector ($G_{s,t}$, V and π). The DA model promotes positive transfer by reducing distribution shift in the shared label space. The RDS based on RL mitigates negative transfer by filtering outlier source classes. Specifically, to filter outlier source samples, the policy network π considers high-level information provided by feature extractor F and classifier C for decision making to get selected source samples $\boldsymbol{X}^{s'}$. For the backbone of DA model, C takes source transfer features $\boldsymbol{Z}^{s'} = F(\boldsymbol{X}^{s'})$ as input to produce label predictions $\hat{Y}^s$, and F achieves distribution alignment between $F(\boldsymbol{X}^{s'})$ and $F(\boldsymbol{X}^t)$. Meanwhile, the selected source samples' reconstruction errors $\|\boldsymbol{X}^{s'} - G_t F(\boldsymbol{X}^{s'})\|_2^2$ based on G_t are used as a reward to encourage π to select samples with small reconstruction errors. For the stability of training, based on actor-critic algorithm, we use a value network V combined with rewards to optimize π. Besides, the domain-specific generators G_s and G_t trained with reconstruction errors of reconstructed source images $G_s(F(\boldsymbol{X}^{s\prime}))$ and target images $G_t(F(\boldsymbol{X}^t))$, respectively.

3.2. Domain Adaptation Model

Almost all PDA frameworks are based on adversarial network [4, 5, 32, 6], which has led to many existing DA algorithms based on moment matching cannot be extended to solve the PDA problem. The proposed RDS is a general module that can be integrated into most DA frameworks. In this work, we use deep CORAL[25] as the base DA model to prove that RDS can be embedded into the matching-based DA framework to make it robust to PDA scene. The reason why we chose CORAL is that it is simple and effective. Although there are some limitations in CORAL, it is beyond our research scope. Besides, the RDS is universal, so CORAL can be replaced with other better DA methods. In the Appendix, we also provide a solution for embedding RDS into DANN to demonstrate that RDS can also be integrated into the method based on adversarial network. In the following, we will give a brief introduction of CORAL.

We define the last layer of F as adaptation layer and reduce the distribution shift between source and target domains by aligning the covariance matrix of source and target features. Hence, the CORAL objective function is:

$$\underset{(F)}{\mathcal{L}_c} = \frac{1}{d^2}\|Cov(\boldsymbol{Z}_b^s) - Cov(\boldsymbol{Z}_b^t)\|_F^2, \tag{1}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm, $\boldsymbol{Z}_b^s \in \mathbb{R}^{n\times d}$ and $\boldsymbol{Z}_b^t \in \mathbb{R}^{n\times d}$ represent source and target transferable features output by the adaptation layer, respectively, b is the batch ID, d is the dimension of the transferable feature, and n is the batch size. $Cov(\boldsymbol{Z}_b^s)$ and $Cov(\boldsymbol{Z}_b^t)$ represent the covariance matrices, which can be computed as $Cov(\boldsymbol{Z}_b^s) = \mathbf{Z}_b^{s\top}\mathbf{Z}_b^s$, and $Cov(\mathbf{Z}_b^t) = \mathbf{Z}_b^{t\top}\mathbf{Z}_b^t$.

To ensure the shared feature extractor and classifier can be trained with supervision on labeled samples, we define a standard cross-entropy classification loss $\mathcal{L}_s$ with respect to labeled source samples. Formally, the full objective function for the domain adaptation model is as follows:

$$\mathcal{L}_{DA} = \mathcal{L}_s + \lambda_1 \mathcal{L}_c, \tag{2}$$

where hyperparameter λ_1 control the impact of the corresponding objective function. However, in the PDA scenario, most DA methods (e.g. CORAL) may trigger negative transfer since these methods force alignment of the global

distributions p and q, even though $p_{\mathcal{C}_s \setminus \mathcal{C}_t}$ and q are non-overlapping and cannot be aligned during transfer. Thus, the motivation of the reinforced data selector is to mitigate negative transfer by filtering out the outlier source classes $\mathcal{C}_s \setminus \mathcal{C}_t$ before performing the distribution alignment.

3.3. Reinforced Data Selector

We consider the source sample selection process of RTNet as Markov decision process, which can be addressed by RL. The RDS is an agent that interacts with the environment created by the DA model. The agent takes action to keep or drop a source sample based on the policy function. The DA model evaluates the actions taken by the agent and provides a reward to guide the learning of agent.

As shown in Figure 2, given a batch of source samples $\boldsymbol{X}_b^s = \{\boldsymbol{x}_i^s\}_{i=1}^n$, we can obtain the corresponding states $\boldsymbol{S}_b^s = \{\boldsymbol{s}_i^s\}_{i=1}^n$ through the DA model. The RDS then utilizes the policy $\pi(\boldsymbol{S}_b^s)$ to determine the actions $\boldsymbol{A}_b^s = \{a_i^s\}_{i=1}^n$ taken on source samples, where $a_i^s \in \{0, 1\}$. $a_i^s = 0$ means to filter outlier sample from $\boldsymbol{X}_b^s$. Thus, we get a new source batch $\boldsymbol{X}_b^{s'}$ related to target domain. Instead of $\boldsymbol{X}_b^s$, we feed $\boldsymbol{X}_b^{s'}$ into the DA model to solve the PDA problem. Finally, the DA model moves to the next state s' after updated with $\boldsymbol{X}_b^{s'}$ and $\boldsymbol{X}_b^t$, and provides a reward r_b according to the source reconstruction errors based on G_t to update π and V. In the following sections, we will give a detailed introduction to the state, action, and reward.

State: State is defined as a vector $\boldsymbol{s}_i^s \in \mathbb{R}^l$. In order to simultaneously consider the unique information of each source sample and the label distribution of target domain when taking action, $\boldsymbol{s}_i^s$ concatenates the following features: (1) The high-level semantic feature $\boldsymbol{z}_i^s$, which is the output of F given $\boldsymbol{x}_i^s$, i.e., $\boldsymbol{z}_i^s = F(\boldsymbol{x}_i^s)$. (2) The label y_i^s of the source sample, represented by a one-hot vector. (3) The predicted probability distribution α of the target batch $\boldsymbol{X}_b^t$, which can be calculated as $\frac{1}{n}\sum_{i=1}^n \hat{y}_i^t$, $\hat{y}_i^t = C(F(x_i^t))$. Feature (1) represents high-level information of source sample. Feature (3) based on the intuition that the probabilities of assigning the target data to outlier source classes should be small since the target sample is significantly dissimilar to the outlier source sample. Consequently, α quantifies the contribution of each source class to the target domain. Feature (2) is combined with feature (3) to measure the relation between each source sample and the target domain.

Action: The action $a \in \{0, 1\}$, which indicates whether the source sample is kept or filtered from the source batch. The selector utilizes ϵ-greedy strategy [19] to sample a based on $\pi(\boldsymbol{s}_i^s)$. $\pi(\boldsymbol{s}_i^s) \in \mathbb{R}^1$ represents the probability that the sample is kept. The ϵ is decayed from 1 to 0 as the training progresses. π is defined as a policy network with two fully connected layers. Formally, $\pi(\boldsymbol{s}_i^s)$ is computed as:

$$\pi(\boldsymbol{s}_i^s) = sigmoid(\boldsymbol{W}_2\delta(\boldsymbol{W}_1\boldsymbol{s}_i^s + \boldsymbol{b}_1) + \boldsymbol{b}_2), \quad (3)$$

where δ is the ReLU activation, $\boldsymbol{W}_k$ and $\boldsymbol{b}_k$ are the weight matrix and bias of the k-th layer, and $\boldsymbol{s}_i^s$ is the state of the source sample, which concatenates feature (1), (2) and (3).

Reward: The selector takes actions to select $\boldsymbol{X}_b^{s'}$ from $\boldsymbol{X}_b^s$. The RTNet uses $\boldsymbol{X}_b^{s'}$ to update the DA model and obtains a reward r_b for evaluating the policy. In contrast to usual reinforcement learning, where one reward corresponds to one action, The RTNet assigns one reward to a batch of actions to improve the efficiency of model training.

To take advantage of pixel-level information when selecting source samples, the novel reward is designed according to the reconstruction error of the selected source sample based on G_t. The intuition of using this reconstruction error as reward is that the reconstruction error $\|\boldsymbol{x}_i^{s'} - G_tF(\boldsymbol{x}_i^{s'})\|_2^2$ of outlier source sample is large since they are extremely dissimilar to the target classes. Hence, the selector aims to select source samples with small reconstruction errors for distribution alignment and classifier training. However, the purpose of RL is to maximize the reward, so we design the following novel reward based on reconstruction error:

$$r_b = \exp(-\frac{1}{n'}\sum_{i=1}^{n'} \|\boldsymbol{x}_i^{s'} - G_tF(\boldsymbol{x}_i^{s'})\|_2^2), \quad (4)$$

where $\boldsymbol{x}_i^{s'}$ is the sample selected by the reinforced data selector, and n' is the number of samples selected. As shown in Eq. 4, the smaller the reconstruction error, the greater the reward, which is in line with our expectations. Note that, to accurately evaluate the efficacy of $\boldsymbol{X}_b^{s'}$, rewards are collected after the feature extractor F and classifier C are updated as in Eq. 5 and before the generators $G_{s,t}$ are updated as in Eq. 6. F, C and $G_{s,t}$ can be trained as follows:

$$\min_{(F,C)} \mathcal{L}_{DA}, \quad (5)$$

$$\min_{(G_{s,t},F)} \frac{1}{n'}\sum_{i=1}^{n'} \|\boldsymbol{x}_i^{s'} - G_sF(\boldsymbol{x}_i^{s'})\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i^t - G_tF(\boldsymbol{x}_i^t)\|_2^2. \quad (6)$$

In the process of selection, not only the last action contributes to the reward, but all previous actions contribute. Therefore, the future total reward r_b' for each batch b can be formalized as:

$$r_b' = \sum_{j=0}^{N-b} \gamma^j r_{b+j}, \quad (7)$$

where γ is the reward discount factor, and N is the number of batches in this episode.

Optimization: The selector is optimized based on actor-critic algorithm [15]. In each episode, the selector aims to maximize the expected total reward. Formally, the objective function is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{b=1}^{N} r_b], \quad (8)$$

where θ is the parameter of policy network π. θ is updated by performing, typically approximate, gradient ascent on $\mathcal{J}(\theta)$. Formally, the update step of π is defined as:

$$\theta = \theta + l * \frac{1}{n}\sum_{i=1}^{n} v_i \nabla_\theta \log(\pi_\theta(\boldsymbol{s}_i^s)), \tag{9}$$

where l is the learning rate, n is the batch size, and v_i is an estimate of the advantage function based on future total reward, which guides the update of π. Note that, $v_i \nabla_\theta \log(\pi_\theta(\boldsymbol{s}_i^s))$ is an unbiased estimate of $\nabla_\theta \mathcal{J}(\theta)$ [30]. The actor-critic framework combines π and V for stable training. In this work, we utilize $V_\Omega(\boldsymbol{s}_i^s)$ to estimate the expected feature total reward. Hence, the v_i can be considered as an estimate of the advantage of action, which can be defined as follows:

$$v_i = r'_b - V_\Omega(\boldsymbol{s}_i^s). \tag{10}$$

The architecture of the value network V is similar to policy network, except that the final output layer is a regression function. V is designed to estimate the expected feature total reward for each state, which can be optimized by:

$$\Omega = \Omega - l * \frac{1}{n}\sum_{i=1}^{n} \nabla_\Omega \|r'_b - V_\Omega(\boldsymbol{s}_i^s)\|_2^2, \tag{11}$$

where Ω is the trainable parameters of value network V.

Algorithm 1 The optimization strategy of the RTNet

Require: episode number L, source data $\{\boldsymbol{X}^s, Y^s\}$ and target data $\boldsymbol{X}^t$.

Initialize each module in the RTNet.
for $episode = 1 \rightarrow L$ **do**
  for each $(\boldsymbol{X}_b^s, Y_b^s), (\boldsymbol{X}_b^t) \in (\boldsymbol{X}^s, Y^s), (\boldsymbol{X}^t)$ **do**
    Obtain the states $\boldsymbol{S}_b^s = \{\boldsymbol{s}_i^s\}_{i=1}^n$ through the domain adaptation model, where $\boldsymbol{s}_i^s = [F(x_i^s), y_i^s, \alpha]$.
    Utilizes ϵ-greedy strategy to sample $A_b^s = \{a_i^s\}_{i=1}^n$ based on $\pi(\boldsymbol{S}_b^s)$.
    Select source training batch $(\boldsymbol{X}_b^{s'}, Y_b^{s'})$ from $(\boldsymbol{X}_b^s, Y_b^s)$ according to A_b^s.
    Update domain adaptation model (F and C) with $(\boldsymbol{X}_b^{s'}, Y_b^{s'})$ and $(\boldsymbol{X}_b^t)$ as in Eq. 5.
    Obtain reward r_b on G_t with $\boldsymbol{X}_b^{s'}$ as in Eq. 4.
    Update $G_{s,t}$ with $\boldsymbol{X}_b^{s'}$ and $\boldsymbol{X}_b^t$ as in Eq. 6.
    Store $(\boldsymbol{S}_b^s, A_b^s, r_b)$ to an episode history H.
  end for
  for each $(\boldsymbol{S}_b^s, A_b^s, r_b) \in H$ **do**
    Obtain the future total reward r'_b as in Eq. 7.
    Obtain the estimated future total reward $V(\boldsymbol{S}_b^s)$.
    Update π as Eq. 9 and update V as Eq. 11.
  end for
end for

As the RDS and DA model interact with each other during training, we train them jointly. To ensure that the DA model can provide accurate states and rewards in the early stages of training, we first pre-train $G_{s,t}$, F, and C through the classification loss $\mathcal{L}_s$ of source samples and Eq. 6. We follow the previous work [22] to train the RTNet, the detailed training process is shown in Algorithm 1.

3.4. Theoretical Analysis

In this section, we prove theoretically that our method improves the expected error boundary on the target sample by using the theory of domain adaptation [2].

Theorem 1. *Let $\mathcal{H}$ be the common hypothesis class for source $\mathcal{S}$ and target $\mathcal{T}$, the upper bound of the expected error for the target domain, $\epsilon_t(h)$, is defined as:*

$$\epsilon_t(h) \leq \epsilon_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(p, q) + C, \forall h \in \mathcal{H}, \tag{12}$$

where the expected error for the target domain is bounded by three terms: (1) $\epsilon_s(h)$ is the expected error for source domain; (2) $d_{\mathcal{H}\Delta\mathcal{H}}(p, q)$ is the domain divergence measured by a discrepancy distance between source distribution p and target distribution q; (3) $C = min_h[\epsilon_s(h) + \epsilon_t(h)]$ is the shared error of the ideal joint hypothesis.

In Eq. 12, $\epsilon_s(h)$ is expected to be small due to it can be optimized by a deep network with the source labels. Prior DA methods [27, 17, 25, 12] seek to minimize $d_{\mathcal{H}\Delta\mathcal{H}}(p, q)$ by aligning the global distributions of $\mathcal{S}$ and $\mathcal{T}$. However, Eq. 12 assumes that the label space of the source and target domains is consistent, which is not held in the PDA scenario. Therefore, blindly aligning the global distribution is an erroneous solution, which forces the target sample to align with the outlier source classes (Figure 1a), resulting in a large $\epsilon_t(h)$ in C and triggering a negative transfer. To this end, we need to ensure the consistency of the source and target label spaces. However, it is not possible to directly filter the outlier source classes as the label space of the target domain is unknown. Hence, we propose the RTNet, which extends the DA methods to automatically filter the outlier source classes, so that the Eq. 12 can get the correct results.

4. Experiments

4.1. Datasets

Office-31[24] is a widely-used visual domain adaptation dataset, which contains 4,110 images of 31 categories from three distinct domains: Amazon website (A), Webcam (W) and DSLR camera (D). Following the settings in [4], we select the same 10 categories in each domain to build new target domains and create 6 transfer scenarios as in Table 1.

Table 1: Performance on Office-31 dataset and Digital Dataset. RTNet$_{adv}$ represents the model that integrates the reinforced data selector into the DANN. For the integrity and readability of the paper, RTNet$_{adv}$ will be introduced in the Appendix.

Type	Method	Office-31							Digital Dataset				
		A31→W10	D31→W10	W31→D10	A31→D10	D31→A10	W31→A10	Avg	SVHN10 → MNIST5	MNIST10 →MNIST-M5	USPS10 →MNIST5	SYN10 →MNIST5	Avg
NoA	ResNet / LeNet	76.5±0.3	99.2±0.2	97.7±0.1	87.5±0.2	87.2±0.1	84.1±0.3	88.7	79.6±0.3	60.2±0.4	76.6±0.6	91.3±0.4	76.9
DA	DAN[17]	53.6±0.7	62.7±0.5	57.8±0.6	47.7±0.5	61.2±0.6	69.7±0.5	58.8	63.5±0.5	48.9±0.5	61.3±0.4	55.0±0.3	57.2
	DANN[12]	62.8±0.6	71.6±0.4	65.6±0.5	65.1±0.7	78.9±0.3	79.2±0.4	70.5	68.9±0.7	50.6±0.7	83.3±0.5	77.6±0.4	70.1
	CORAL[25]	52.1±0.5	65.2±0.2	64.1±0.7	58.0±0.5	73.1±0.4	77.9±0.3	65.1	60.8±0.6	43.4±0.5	61.7±0.5	74.4±0.4	60.1
	JDDA[7]	73.5±0.6	93.1±0.3	89.3±0.2	76.4±0.4	77.6±0.1	82.8±0.2	82.1	72.1±0.4	54.3±0.2	71.7±0.4	85.2±0.2	70.8
PDA	PADA [5]	86.3±0.4	99.3±0.1	**100**±0.0	90.4±0.1	91.3±0.2	92.6±0.1	93.3	90.4±0.3	89.1±0.2	97.4±0.3	96.5±0.1	93.4
	ETN[6]	93.4±0.3	99.3±0.1	99.2±0.2	95.5±0.4	**95.4**±0.1	91.7±0.2	95.8	93.6±0.2	92.5±0.1	96.5±0.1	97.8±0.2	95.1
	RTNet	95.1±0.3	**100**±0.0	**100**±0.0	**97.8**±0.1	93.9±0.1	94.1±0.1	96.8	95.3±0.1	94.2±0.2	**98.9**±0.1	99.2±0.0	96.9
	RTNet$_{adv}$	**96.2**±0.3	**100**±0.0	**100**±0.0	97.6±0.1	92.3±0.1	**95.4**±0.1	**96.9**	**97.2**±0.1	**94.6**±0.2	98.5±0.1	**99.7**±0.0	**97.5**

Table 2: Performance on the Office-Home dataset. RTNet$_{adv}$ represents the model that integrates the reinforced data selector into the DANN. For the integrity and readability of the paper, RTNet$_{adv}$ will be introduced in the Appendix.

Type	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
NoA	ResNet	47.2±0.2	66.8±0.3	76.9±0.5	57.6±0.2	58.4±0.1	62.5±0.3	59.4±0.3	40.6±0.2	75.9±0.3	65.6±0.1	49.1±0.2	75.8±0.4	61.3
DA	DAN[17]	35.7±0.2	52.9±0.4	63.7±0.2	45.0±0.3	51.7±0.3	49.3±0.1	42.4±0.2	31.5±0.4	68.7±0.1	59.7±0.3	34.6±0.4	67.8±0.1	50.3
	DANN[12]	43.2±0.5	61.9±0.2	72.1±0.4	52.3±0.4	53.5±0.2	57.9±0.1	47.2±0.3	35.4±0.1	70.1±0.3	61.3±0.2	37.0±0.2	71.7±0.3	55.3
	CORAL[25]	38.2±0.1	55.6±0.3	65.9±0.2	48.4±0.4	52.5±0.1	51.3±0.2	48.9±0.3	32.6±0.1	67.1±0.2	63.8±0.4	35.9±0.2	69.8±0.1	52.5
	JDDA[7]	45.8±0.4	63.9±0.2	74.1±0.3	51.8±0.2	55.2±0.3	60.3±0.2	53.7±0.2	38.3±0.1	72.6±0.2	62.5±0.1	43.3±0.3	71.3±0.1	57.7
PDA	PADA[5]	53.2±0.2	69.5±0.1	78.6±0.1	61.7±0.2	62.7±0.3	60.9±0.1	56.4±0.5	44.6±0.2	79.3±0.1	74.2±0.1	55.1±0.3	77.4±0.2	64.5
	ETN[6]	60.4±0.3	76.5±0.2	77.2±0.3	64.3±0.1	67.5±0.3	75.8±0.2	69.3±0.1	54.2±0.1	83.7±0.2	75.6±0.3	56.7±0.2	84.5±0.3	70.5
	RTNet	62.7±0.1	79.3±0.2	**81.2**±0.1	65.1±0.1	68.4±0.3	76.5±0.1	70.8±0.2	**55.3**±0.1	**85.2**±0.3	76.9±0.2	**59.1**±0.2	83.4±0.3	72.0
	RTNet$_{adv}$	**63.2**±0.1	**80.1**±0.2	80.7±0.1	**66.7**±0.1	**69.3**±0.2	**77.2**±0.2	**71.6**±0.3	53.9±0.3	84.6±0.1	**77.4**±0.2	57.9±0.3	**85.5**±0.1	**72.3**

Digital Dataset includes five domain adaptation benchmarks: Street View House Numbers (SVHN) [20], MNIST [16], MNIST-M [12], USPS [14] and synthetic digits dataset (SYN) [12], which consist of ten categories. We select 5 categories (digit 0 to digit 4) as target domain in each dataset and construct four PDA tasks as in Table 1.

Office-Home[28] is a more challenging DA dataset, which consists of 4 different domains: Artistic images (Ar), Clipart images (Cl), Product images (Pr) and Real-World images (Rw). For each transfer task, when a domain is used as the source domain, we use samples from all 65 categories; when a domain is used as the target domain, we select the samples from the same 25 categories as [6]. Hence, we can build twelve PDA tasks as in Table 2.

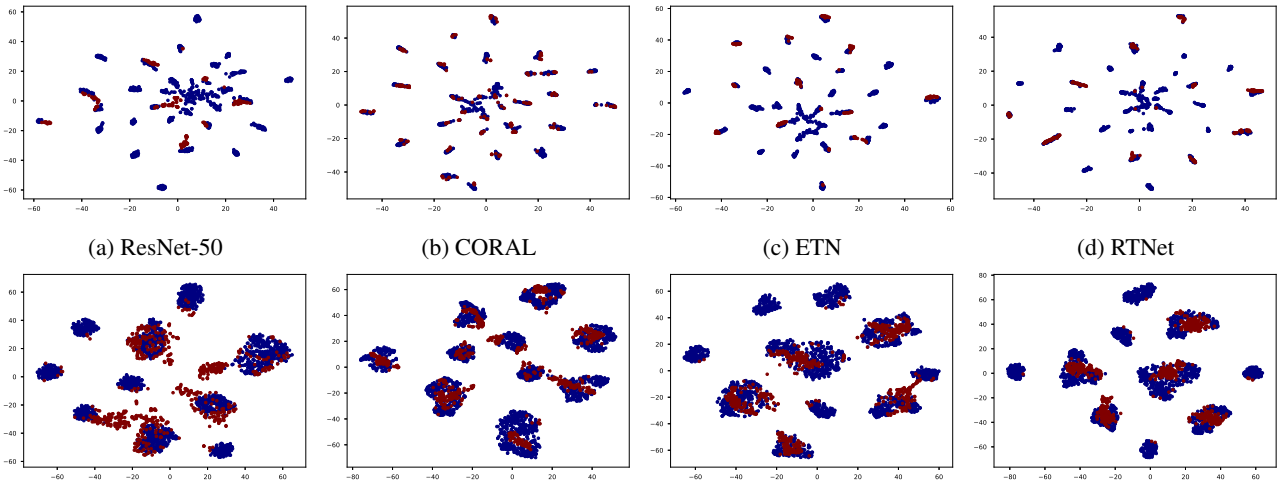
4.2. Implementation Details

The RTNet is implemented via Tensorflow and trained with the Adam optimizer. For the experiments on Office-31 and Office-Home, we employ the ResNet-50 pre-trained on ImageNet as the backbone of domain adaptation model and fine-tune the parameters of the fully connected layers and the final block. For the experiments on digital datasets, we adopt modified LeNet as the backbone of domain adaptation model and update all of the weights. All images are converted to grayscale and resized to 32×32.

In RTNet, the structure of each module can be seen in Appendix. To guarantee fair comparison, the same frameworks are used for F and C in all comparison methods, and each method is trained five times and the average is taken as the final result. For all hyperparameters, we set $l = 1e-4$, $\lambda_1 = 7$ and $\gamma = 0.85$, which selected by using a grid search on the performance of the validation set. The parameter sensitivity analysis can be seen in Appendix. To ease model selection, the hyperparameters of comparison methods are gradually changing from 0 to 1 as in [18].

4.3. Result and Discussion

Tables 1 and 2 show the classification results on three datasets. By looking these tables, several observations can be made. (1) the previous standard DA methods including those based on adversarial network (DANN), and those based on moment match (DAN, JDDA, and CORAL) perform even worse than non-adaptation (NoA) model, indicating that they were affected by the negative transfer. (2) PDA methods (ETN and PADA) improve classification accuracy by a large margin since their weighting mechanisms can mitigate negative transfer caused by outlier categories. (3) Comparing the model with the RDS (RTNet and RTNet$_{adv}$) and the model without the RDS (CORAL and DANN), the model with the RDS can alleviate the negative transfer to greatly improve the performance of the model in the target domain. This proves that the selector we design is a general model and can be easily integrated into existing DA models, including not only matching-based methods but also adversarial-based methods. (4) RTNet / RTNet$_{adv}$ achieves the best accuracy on most transfer tasks. Different from



(a) ResNet-50 (b) CORAL (c) ETN (d) RTNet

(e) LeNet (f) CORAL (g) ETN (h) RTNet

Figure 3: The t-SNE visualization on A31→W10 ((a)-(d)) and SVHN10→MNIST5 ((e)-(h)). Red points represent target samples and blue points represent source samples. The results generated from category information are shown in Appendix.

the previous PDA methods which only rely on the high-level information to obtain the weight, RTNet / RTNet$_{adv}$ adopts the high-level information to select the source sample, and employs the pixel-level information as evaluation criteria to guide the learning of policy network. Thus, this selection mechanism can detect outlier source classes more effectively and transfer relevant samples.

4.4. Analysis

Feature Visualization: We visualize the features of adaptation layer using t-SNE [10]. As shown in Figure 3, several observations can be made. (1) by comparing Figures 3a, 3e and Figures 3b, 3f, we find that CORAL forces the target domain to be aligned with the whole source domain, including outlier classes that do not exist in the target label space, which triggers negative transfer. (2) as can be seen in Figures 3d, 3h, RTNet correctly matches the target samples to related source samples by integrating the RDS into CORAL to filter outlier classes, which confirms that the matching-based DA methods can be extended to solve PDA problem by embedding RDS. (3) compared with Figure 3c, 3g, RTNet matches the related source domain and the target domain more accurately, indicating that it is more effective than ETN in suppressing the side effect of outlier classes by considering high-level and pixel-level information.

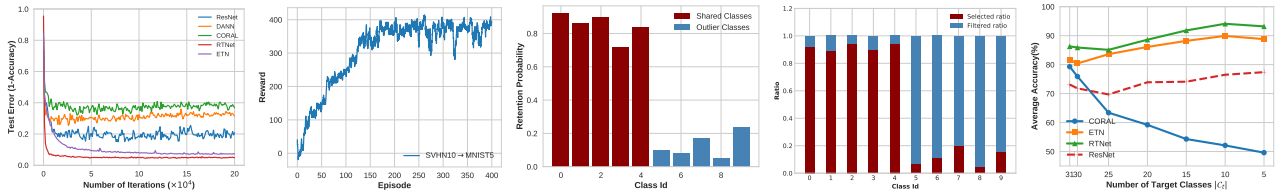
Convergence Performance: We analyze the convergence of RTNet. As shown in Figure 4a, the test errors of DANN and CORAL are higher than ResNet due to negative transfer. RTNet fast and stably converges to the lowest test error, indicating that it can be efficiently trained to solve PDA problem. As shown in Figure 4b, the reward gradually increases as the episode progresses, meaning that the RDS can learn the correct policy to maximize the reward and filter out outlier source classes.

4.5. Case Study and Performance Interpretation

The results in Section 4.3 demonstrate the effectiveness of the RTNet. However, the lack of interpretability of the neural architecture makes it difficult to speculate on the reasons behind decisions made by RDS. Therefore, we introduce the overall performance and specific case to prove the ability of the selector to select and filter samples.

Statistics of Class-wise Retention Probabilities: We utilize $E(\pi_\theta(\boldsymbol{S}_c^s))$ to verify the ability of the selector to filter the samples, averaging the retention probabilities of each class of source domain. $\boldsymbol{S}_c^s$ represents a source sample set, which contains samples belonging to class c. As shown in Figure 4c, RTNet assigns much larger retention probabilities to the shared classes than to the outlier classes. These results prove that RTNet has the ability to automatically select relevant source classes and filter out outlier classes. Besides, the outlier classes with a similar appearance to the shared classes, such as 7 and 9, have a larger retention probability than other outlier classes, which indicates that the selector we develop can indeed select source samples similar to the target domain based on the pixel-level information.

Class-wise Selected Ratio and Filtered Ratio: We input the sampled SVHN samples into RTNet for sample selection. As shown in Figure 4d, the outlier samples (5, 6 and 8) that differ significantly in appearance from the shared samples (0-4) can be filtered out by 92% on average, while the outlier classes (7 and 9) with smaller appearance differences from the shared classes can be filtered out by 72% on average. For shared classes, the ratio of samples filtered



(a) Convergence (b) Learning curve (c) Probability in $\mathcal{C}_s$ (d) Select & filter ratio (e) Acc w.r.t. $|\mathcal{C}_t|$

Figure 4: (a) Convergence analysis on SVHN10→MNIST5. (b) Learning curve on SVHN10→MNIST5. (c) Source class-wise retention probability learned by policy network on SVHN10→MNIST5. (d) Class-wise selected ratio and filtered ratio evaluated by RDS trained on SVHN10→MNIST5. (e) The accuracy curve of varying the number of target classes on A→W.

in each class is not much different, and an average of 8.2% of the samples are filtered by error. These results indicate that RTNet can effectively filter outlier source samples, especially those that have large differences in appearance with shared classes and keep related source samples well.

Wasserstein Distances between Domains: The Wasserstein distance measures the distance between two probability distributions [29]. We take the selection of the selector in the last episode as the result of the selection and calculate the Wasserstein distance between target samples and source samples, including selected and filtered source samples. We observe that the patterns of the two tasks are identical through the results of Table 3: (1) $W_{select} < W_{all}$, which indicates that the sample selected by the selector is closer to the target domain and thus may contribute to the transfer process. (2) $W_{filter} > W_{all}$, which means that the filtered source samples are extremely dissimilar to the target domain and may result in a negative transfer. These findings show that our proposal can select source samples whose Wasserstein distances are close to the target domain. This makes sense because such source samples can be more easily transferred and helpful to the target domain.

Table 3: The Wasserstein distances between domains.

Name	Domains	SVHN10 →MNIST5	A31→W10
W_{all}	T ↔ S	0.2574	4.3233
W_{select}	T ↔ S_{select}	0.1645	2.3179
W_{filter}	T ↔ S_{filter}	0.3679	5.6308

Target Classes: We conduct experiments to evaluate the performance of RTNet when the number of target classes varies. As shown in Figure 4e, as the number of target classes decreases, the performance of CORAL degrades rapidly, indicating that negative transfer becomes more and more serious as the label distribution becomes larger. RTNet performs better than other methods, indicating that it can suppress negative transfer more effectively. Moreover, RTNet is superior to CORAL when the source and target label spaces are consistent (A31→W31), which shows that our method does not filter erroneously when there are no outlier classes. As shown in Figure 5, on the A31→W31 task, we analyze some of the source samples filtered by RDS and find that most of them are noise samples, which have mismatches between labels and images. For example, an image of a mouse is wrongly labeled as a keyboard in the Office-31 dataset. This case study shows that the RDS can filter noisy samples to improve the performance even if the label space of the source and target domains are consistent.

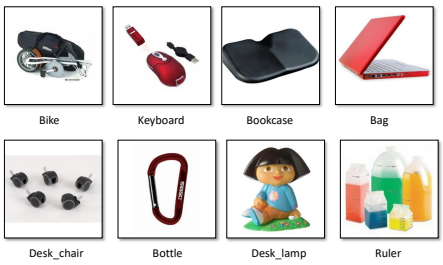


Figure 5: Case study on A31→W31 task. These noisy samples with mismatches between labels and images are sampled from the source samples filtered by RDS. The description below the image is the label provided by the dataset.

5. Conclusion

In this work, we propose an end-to-end RTNet, which utilizes both high-level and pixel-level information to address PDA problem. RTNet applies RL to train a reinforced data selector based on actor-critic framework to filter outlier source classes with the purpose of mitigating negative transfer. Unlike previous adversarial-based PDA methods, The RDS we proposed can be integrated into almost all DA models including those based on adversarial network, and those based on moment match. Note that, the results of RTNet$_{adv}$ based on adversarial model are shown in Appendix. The state-of-the-art experimental results confirm the efficacy of RTNet.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

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

[3] J Quiñonero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning, 2009.

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *CVPR*, pages 2724–2732, 2018.

[5] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *ECCV*, pages 135–150, 2018.

[6] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. *arXiv preprint arXiv:1903.12230*, 2019.

[7] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI*, volume 33, pages 3296–3303, 2019.

[8] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. *arXiv preprint arXiv:1912.11976*, 2019.

[9] Zhihong Chen, Chao Chen, Xinyu Jin, Yifu Liu, and Zhaowei Cheng. Deep joint two-stream wasserstein auto-encoder and selective attention alignment for unsupervised domain adaptation. *Neural Computing and Applications*, pages 1–14, 2019.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[11] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In *EMNLP*, pages 595–605, 2017.

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

[13] 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*, 2018.

[14] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 16(5):550–554, 2002.

[15] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *NeurIPS*, pages 1008–1014, 2000.

[16] Y. L. Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. proc ieee. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[20] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning & Unsupervised Feature Learning*, 2011.

[21] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[22] Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W Bruce Croft. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *WSDM*, pages 699–707, 2019.

[23] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, England, 1994.

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

[25] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450, 2016.

[26] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, volume 1, page 4, 2017.

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

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

[29] Cédric Villani. Topics in optimal transportation. *Ams Graduate Studies in Mathematics*, page 370, 2003.

[30] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[31] Jiawei Wu, Lei Li, and William Yang Wang. Reinforced co-training. In *NAACL HLT 2018*, pages 1252–1262, 2018.

[32] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, pages 8156–8164, 2018.