Reinforced Adaptation Network for Partial Domain Adaptation

Keyu Wu[1], Min Wu[1], Zhenghua Chen[1,2*], Ruibing Jin[1], Wei Cui[1], Zhiguang Cao[1] and Xiaoli Li[1,2]

***Abstract*—Domain adaptation enables generalized learning in new environments by transferring knowledge from label-rich source domains to label-scarce target domains. As a more realistic extension, partial domain adaptation (PDA) relaxes the assumption of fully shared label space, and instead deals with the scenario where the target label space is a subset of the source label space. In this paper, we propose a Reinforced Adaptation Network (RAN) to address the challenging PDA problem. Specifically, a deep reinforcement learning model is proposed to learn source data selection policies. Meanwhile, a domain adaptation model is presented to simultaneously determine rewards and learn domain-invariant feature representations. By combining reinforcement learning and domain adaptation techniques, the proposed network alleviates negative transfer by automatically filtering out less relevant source data and promotes positive transfer by minimizing the distribution discrepancy across domains. Experiments on three benchmark datasets demonstrate that RAN consistently outperforms seventeen existing state-of-the-art methods by a large margin.**

***Index Terms*—Deep reinforcement learning; partial domain adaptation; domain adaptation; transfer learning**

I. Introduction

DEEP neural networks have revolutionized the fields of machine learning and achieved unprecedented performance in a variety of applications. Generally, the availability of large-scale labelled data is an important prerequisite for their significant advances. However, in practice, it is unrealistic to collect sufficient labelled data for every new application considering the prohibitive cost of data annotation. Meanwhile, it is also infeasible to apply existing well-trained models directly to new domains due to the severe degradation in generalization ability resulted from domain shift [1]–[3]. Since domain adaptation enables knowledge transfer from label-rich source domains to unlabelled target domains through reducing the distribution discrepancy, it plays an important role to alleviate the burden of labelling.

Most existing domain adaptation (DA) methods attempt to learn domain-invariant feature representations by embedding distribution matching modules into the network architectures [4], [5]. For instance, a number of unsupervised domain adaptation methods diminish domain discrepancy by incorporating a specific distribution similarity measure, such as Maximum Mean Discrepancy [6]–[8], into the network and minimize it along with the standard source classification loss. In contrast, an adversarial domain adaptation approach aligns domain distributions through minimizing an approximate discrepancy in an adversarial training setting [9]–[12].

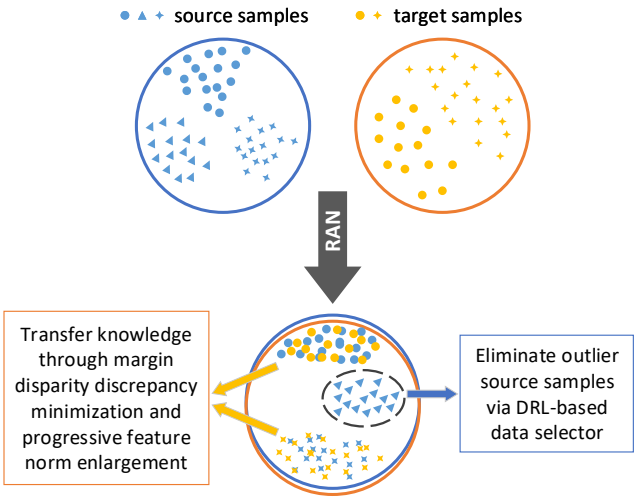


Fig. 1: The objective of the proposed DRL-based partial domain adaptation method is to promote knowledge transfer from source domain to target domain via eliminating outlier source instances and mitigating the domain shift. The DRL model aims to eliminate outlier source samples while the DA model aims to mitigate the domain shift between the target and the selected source samples.

Nevertheless, these domain adaptation algorithms assume that label spaces across domains are identical. However, the label spaces are commonly not fully shared. For example, it is unreasonable to assume that all classification tasks share the same label space as the ImageNet dataset in practice. Due to mismatched label spaces, direct alignment of the whole source domain with the target domain will most likely lead to negative transfer [13]–[15]. Since large-scale labelled datasets are readily accessible as source domain data, a more realistic scenario is partial domain adaptation (PDA) which relaxes the constraint of shared label spaces and assumes that the target label space is a subset of the source label space. As a result, the core of PDA is to recognize relevant source domain data that can promote positive knowledge transfer. So far, several pioneering partial domain adaptation methods have been proposed to address this challenge by up-weighting relevant source instances while down-weighting irrelevant source samples in domain adversarial networks [16]–[18].

Recently, deep reinforcement learning (DRL) algorithms

* Corresponding Author
[1] Institute for Infocomm Research, A*STAR, Singapore, e-mail: (wu_keyu@i2r.a-star.edu.sg; chen0832@e.ntu.edu.sg).
[2] Institute of Information Research and Centre for Frontier AI Research, A*STAR, Singapore.

have also been applied in the context of PDA to learn source data selection policies. In [19], a Reinforced Transfer Network (RTNet) is proposed to eliminate outlier source samples. However, since RTNet utilizes an on-policy DRL algorithm, it can suffer from poor sample efficiency. Besides, it also increases the complexity by introducing additional generator modules to calculate rewards based on the reconstruction errors. In [20], a Domain Adversarial Reinforcement Learning (DARL) framework is introduced to learn policies for the selection of source instances in the shared classes. Nevertheless, DARL relies on a specific domain adversarial learning method to calculate rewards instead of being a general framework that can be integrated into different types of domain adaptation frameworks.

In this paper, we propose a novel DRL-based partial domain adaptation method named Reinforced Adaptation Network (RAN) to address the limitations of the state-of-the-art approaches. As depicted in Fig. 1, RAN jointly train a DRL-based data selector through Double Deep Q-Network (DDQN), and a domain adaptation model through minimizing margin disparity discrepancy and enlarging feature norms of the target and selected source samples. The DRL model is developed to evaluate the transferability of source instances and thereby to automatically eliminate outlier samples that can trigger intrinsic negative transfer. It is trained to output Q-values with similarities between source and target feature representations as rewards. Based on the estimated Q-values, the optimal actions can be determined to either keep or discard source instances. Meanwhile, the domain adaptation model is trained using the selected source data to ultimately yield an adaptive classifier that can be applied to the target domain.

Different from conventional PDA algorithms, RAN better promotes knowledge transfer via combining reinforcement learning and domain adaptation. Meanwhile, it also overcomes the limitations of existing DRL-based PDA methods. First, RAN improves sample efficiency through adopting an off-policy DRL algorithm. In addition, its reward function is also meticulously designed to achieve better performance without adding additional network blocks. Moreover, the DRL paradigm of RAN is general and can be integrated into other unsupervised domain adaptation frameworks as well.

The main contributions of our work can be summarized as follows:

- We have proposed a novel DRL-based partial domain adaptation method to enhance knowledge transfer in partial domain adaptation.
- An innovative framework is proposed to jointly train a DRL model and a domain adaptation model in an efficient way. The domain adaptation model is optimized based on the weights yielded by the DRL model and better learns domain-invariant feature representations via combining margin disparity discrepancy minimization and feature norm enlargement. Meanwhile, the DRL model adapts DDQN to learn source data selection policies based on the rewards provided by the domain adaptation model. These two models can thus leverage each other's strength to achieve better knowledge transfer across domains.
- Extensive experiments have demonstrated that our proposed method consistently outperforms seventeen state-of-the-art approaches by a large margin across three different benchmark datasets.

II. Related Work

Domain Adaptation Domain adaptation aims to bridge domains of different distributions and its key challenge is to mitigate the distribution shift across different domains [13]. In this paper, we focus on unsupervised DA because it is more appealing from the practical application point of view. Recent works have combined unsupervised DA with deep neural networks via adding adaptation layers and minimizing statistical discrepancies between the source and target domains. For instance, Maximum Mean Discrepancy (MMD) [21] based methods minimize the MMD between domains to match their kernel embeddings of distributions [6]–[8], [22]. Inspired by generative adversarial networks (GANs), another category of DA methods introduces a domain classifier subnetwork to learn transferable features in an adversarial manner. For example, the Reverse Gradient method introduced in [23] regards domain discrimination as a binary classification problem and aligns features via reverse gradient backpropagation. Instead, the Adversarial Discriminative Domain Adaptation approach proposed in [2] pre-trains a source encoder and facilitates the domain confusion by training a separate target encoder through a domain-adversarial loss.

Partial Domain Adaptation Partial domain adaptation assumes that the target label space is a subset of the source label space. To solve PDA problems, Selective Adversarial Network (SAN) [16] adopts multiple discriminators to achieve fine-grained adaptation. It weights instances based on their class probabilities to suppress the influence of outlier source classes. Partial Adversarial Domain Adaptation (PADA) [17] extends SAN by employing only one domain adversarial network. It computes class probability of target data predicted by the source classifier and adds this class-level weight to the source classifier. Importance Weighted Adversarial Nets (IWAN) [18] introduces two domain classifiers. It uses the output of an auxiliary domain classifier to predict the probabilities of source instances belonging to the target domain. These domain scores are then used to weight the source examples. Example Transfer Network (ETN) [24] quantifies the weights of source instances based on their similarities calculated by a discriminative domain discriminator, and down-weights outlier examples for the update of source classifier. Stepwise Adaptive Feature Norm (SAFN) [25] enlarges feature norms of the two domains to alleviate negative transfer. Deep Residual Correction Network (DRCN) plugs one residual correction block into the network to mitigate domain discrepancy while aligning target data with the most relevant source subclasses based on a weighted class-wise matching scheme [26]. The dual alignment approach for PDA (DAPDA) [27] proposes a reweighting network to provide class-level weights to source features and a dual alignment network to align both intra-domain and inter-domain distributions. Discriminative Cross-Domain Feature Learning (DCDF) [28] designs a weighted cross-domain center loss and

a weighted cross-domain graph to couple target data to relevant source samples. Adaptive Graph Adversarial Networks (AGAN) [29] realizes PDA through structure-aware domain alignments which utilizes intra-domain and inter-domain edges to construct and train graph neural networks. Multiple Self-Attention Networks (MSAN) [30] decreases domain shift through an attention guided neural network for PDA which is able to learn more fine-grained features in a gradual feature enhancement manner.

Recently, RL techniques have also been used in partial domain adaptation for source data selection. In specific, Domain Adversarial Reinforcement Learning (DARL) [20] uses DQN to automatically learn the data selection policies with domain adversarial learning based rewards. Reinforced transfer network (RTNet) [19] combines both high-level and pixel-level information, and develops a reinforced data selector to filter out outlier source classes. These two pioneering approaches imply the viability of applying reinforcement learning in partial domain adaptation tasks. In this paper, we propose a novel DRL-based partial domain adaptation method to achieve dramatical performance gains over existing methods.

III. Method

In partial domain adaptation, the source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ has n_s labeled examples associated with $|\mathcal{C}_s|$ classes, and the target domain $\mathcal{D}_t = \left\{\mathbf{x}_j^t\right\}_{j=1}^{n_t}$ has n_t unlabelled examples associated with $|\mathcal{C}_t|$ classes. The target domain label space $\mathcal{C}_t$ is a subspace of the source domain label space $\mathcal{C}_s$, i.e. $\mathcal{C}_t \subset \mathcal{C}_s$. The source and target domains are drawn from two different probability distributions, p and q, respectively. Due to the domain shift, $p \neq q$ and $p_{\mathcal{C}_t} \neq q$, where $p_{\mathcal{C}_t}$ denotes the distribution of the source samples in the target label space. Since the target domain is fully unlabelled and its label space $\mathcal{C}_t$ is also unknown, negative transfer can happen if distributions p and q are matched directly. Consequently, filtering out irrelevant source samples is crucial to mitigate negative transfer in PDA. In the meantime, similar to domain adaptation, the distribution shift between $p_{\mathcal{C}_t}$ and q needs to be reduced to promote knowledge transfer from source domain to target domain.

To address the aforementioned challenges, we propose a versatile DRL-based PDA method. As illustrated in Fig. 2, RAN consists of two key components, i.e., a DRL-based data selector and a domain adaptation model. The data selector aims to mitigate negative transfer via eliminating irrelevant source instances. It learns the data selection policies automatically, and determines the optimal actions based on the input high-level features. Meanwhile, the domain adaptation model trains an adaptive classifier based on the selected source samples and simultaneously aligns features from source and target domains by minimizing margin disparity discrepancy and progressively enlarging feature norms of the target and selected source samples.

A. DRL-based Data Selector

We consider the source data selection task as a Markov Decision Process (MDP) which is defined by a tuple $M = (S, A, R, P, \gamma)$, where S and A denote the state and action spaces, respectively, and R, P, γ indicate the immediate reward, state transition function, and discount factor, respectively. At each time step t, the DRL agent executes an action $a_t \in A$ based on its current state $s_t \in S$, subsequently transits to a new state s_{t+1}, and receives a reward $R(s_t, a_t)$. In an MDP, a policy $\pi(a|s)$ specifies the mapping from a state s to an action a and its superiority can be assessed by the Q-value function defined as:

$$Q^\pi(s,a) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0 = s, a_0 = a\right], \quad (1)$$

which is the expectation of discounted sum of rewards, given that action a is taken in state s and policy π is thereafter followed. The objective of the agent is to maximize the expected cumulative future reward. This can be solved by the Q-learning algorithm which approximates the optimal Q-value function iteratively using the Bellman equation shown in the following:

$$Q^*(s_t,a_t) = R(s_t,a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}). \quad (2)$$

The optimal policy can thereby be derived by choosing the action leading to the maximum Q-value, that is, $\pi^*(s) = \underset{a}{\text{argmax}} Q^*(s,a)$. In this paper, we adapt Double Deep Q-Network [31] to determine the optimal data selection policies. Given a batch of source samples $\{\mathbf{x}_i^s\}_{i=1}^{n_b} = \mathbf{X}_b^s$, where b and n_b denote batch ID and batch size respectively, the objective is to obtain a series of actions $\{a_i^s\}_{i=1}^{n_b} = A_b^s$, which in turn determines the weights of the corresponding source samples while updating the network parameters of the domain adaptation module. In the meantime, the rewards $\{R_i^s\}_{i=1}^{n_b}$ are calculated based on the transferability of the source samples. The state, action, reward, and training of our DDQN model are introduced in the following.

State At each training step, a batch of n_b source samples is fed into the feature extractor F of the domain adaptation model to output n_b features $F(\mathbf{X}_b^s) = [F(\mathbf{x}_1^s), \cdots, F(\mathbf{x}_{n_b}^s)]$, where $F(\mathbf{x}_i^s)$ represents the feature vector extracted from source instance $\mathbf{x}_i^s$. Similarly, n_b target features $F(\mathbf{X}_b^t) = [F(\mathbf{x}_1^t), \cdots, F(\mathbf{x}_{n_b}^t)]$ are also obtained. In RAN, a state is defined as the combination of one source feature vector $F(\mathbf{x}_i^s)$ and two additional values which quantify the transferability of the source instance. The dissimilarity between a source sample and the target instances is measured based on the cosine distance and is calculated as:

$$D_i = \min_{F(\mathbf{x}_j^t) \in F(\mathbf{X}_b^t)} 1 - \frac{F(\mathbf{x}_i^s)\cdot F(\mathbf{x}_j^t)}{\|F(\mathbf{x}_i^s)\|_2 \left\|F(\mathbf{x}_j^t)\right\|_2}. \quad (3)$$

Based on this definition, the average dissimilarity of all samples in the source batch can be formulated as:

$$D_{all} = \frac{1}{n_b}\sum_{k=1}^{n_b} D_k. \quad (4)$$

Similarly, the average dissimilarity of all previously selected samples in the source batch before the i-th selection can be

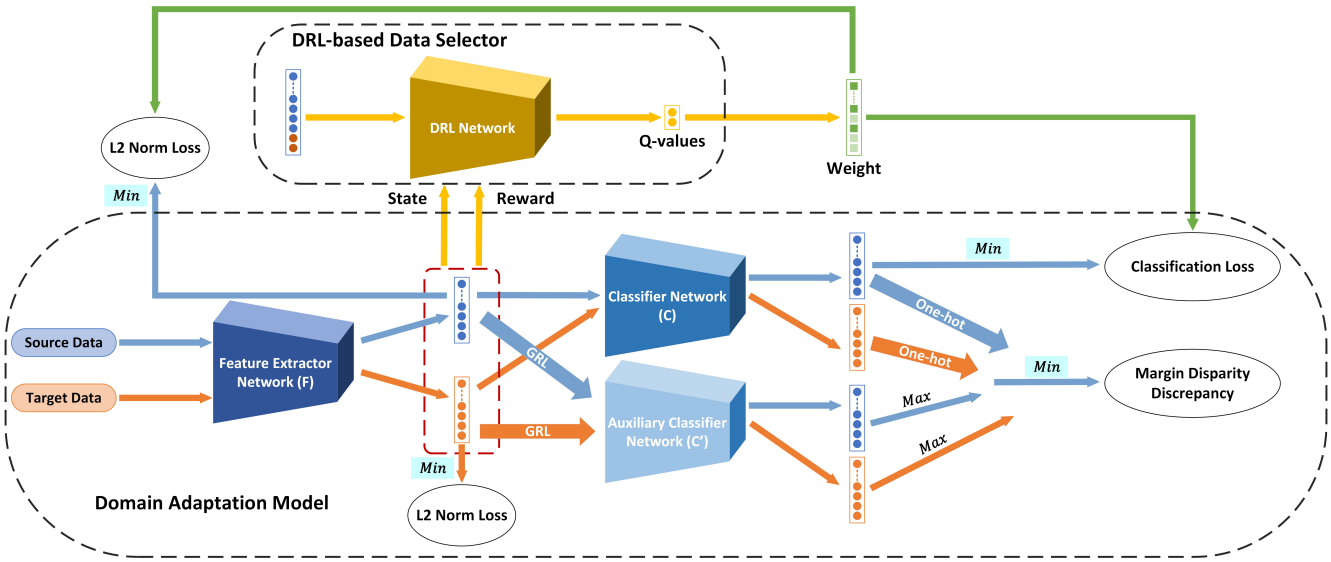


Fig. 2: The proposed DRL-based partial domain adaptation method. It jointly trains a DRL-based data selector and a domain adaptation model. The DRL network automatically eliminates irrelevant source samples based on the estimated Q-values and the rewards are determined based on the relevance of source instances to the target domain. In the meantime, the domain adaptation model is trained based on target and selected source data to learn domain-invariant representations via minimizing classification loss, margin disparity discrepancy, and progressively enlarging feature norm (minimizing L2 norm loss).

formulated as:

$$D_{select} = \frac{\sum_{k=1}^{i-1} D_k I_k}{\sum_{k=1}^{i-1} I_k}, \tag{5}$$

where I_k is a 0-1 vector which indicates whether the k-th source sample is selected. D_{select} is a dynamic value affected by an action to help describe the current situation. Hence, to articulate the transferability of the i-th source sample, the two additional values in the state vector are then defined as $\frac{D_i}{D_{all}}$ and $\frac{D_{select}}{D_{all}}$, respectively. Therefore, a state of DDQN can be expressed as:

$$S(\mathbf{x}_i^s) = \left[F(\mathbf{x}_i^s), \frac{D_i}{D_{all}}, \frac{D_{select}}{D_{all}} \right]. \tag{6}$$

In this way, all source samples in the batch are investigated in order and a terminate state will be triggered at the end. As a result, each batch of source samples can generate n_b experiences for training the DDQN model.

Action The action space of the proposed DDQN model is binary so that each action $a_i \in \{0, 1\}$ denotes whether to keep or discard a source instance. In specific, $a_i = 1$ indicates to select source sample $\mathbf{x}_i^s$ while $a_i = 0$ means to eliminate it. The output of the DDQN model is a two-dimensional vector to represent the Q-values of the binary actions and the optimal action to be taken by the agent at state $S(\mathbf{x}_i^s)$ is determined according to:

$$a_i^* = \underset{a}{\text{argmax}} Q(S(\mathbf{x}_i^s), a). \tag{7}$$

Reward The reward function is to provide feedback on the implementation of the corresponding action and thereby guide the agent to update its data selection policy. Since the objective is to select source data that are more relevant to the target domain, the reward signals are also computed based on the transferability of source examples quantified based on feature similarity. Based on the dissimilarity measure defined in Eq. 3, the reward of taking an action a_i is designed as:

$$R(S(\mathbf{x}_i^s), a_i^s) = 2(A \oplus B - 0.5), \tag{8}$$

where $A = (a_i^s == 1)$ and $B = (D_i \geq D_{all})$ are two boolean functions, and $\oplus$ denotes the exclusive-or operation. That is, a positive reward will be obtained if a source instance is selected while it exhibits higher relevance to the target domain. Meanwhile, a positive reward will also be given if a source instance is filtered out while it demonstrates lower relevance to the target domain. On the contrary, a negative reward will be triggered if a source sample with lower relevance is selected or if a source sample with higher relevance is eliminated.

Training To train the DDQN model, two networks are maintained, including an online network with parameters θ and a target network with parameters θ^-. During training, the actions are selected based on the ϵ-greedy strategy. At each training step, a batch of n_b transitions is sampled from the replay buffer and the estimated optimal Q-value is then updated by minimizing the Huber loss between the predicted Q-value $Q(s_t, a_t; \theta)$ and the target Q-value defined as:

$$Q_t = R(s_t, a_t) + \gamma Q(s_{t+1}, \underset{a_{t+1}}{\text{argmax}} Q(s_{t+1}, a_{t+1}; \theta); \theta^-). \tag{9}$$

The parameters of the online network are updated constantly while those of the target network are softly updated by polyak averaging to generate stable temporal-difference targets.

B. Domain Adaptation Model

The objective of the domain adaptation model is to promote positive transfer through mitigating domain shift. In RAN, this

is achieved by learning an adaptive classifier through minimizing margin disparity discrepancy and enlarging feature norms of the target and selected source samples in a progressive manner. The domain adaptation model is composed of three key components, i.e. a feature extractor F, a classifier C and an auxiliary classifier C'. The optimization of the proposed domain adaptation model consists of three terms: the source classification loss L_y, the margin disparity discrepancy L_d and the feature norm discrepancy L_s. Denote the softmax function σ as:

$$\sigma_i(\mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad \text{for } i = 1, \ldots, m, \tag{10}$$

where $\mathbf{z} = (z_1, \ldots, z_k) \in \mathbb{R}^m$, the classification loss refers to the cross entropy loss defined as:

$$L_y = -\frac{1}{n_s} \sum_{i=1}^{n_s} w(\mathbf{x}_i^s) \log[\sigma_{\mathbf{y}_i^s}(C(F(\mathbf{x}_i^s)))], \tag{11}$$

where $w(\mathbf{x}_i^s) \in \{0, 1\}$ is the binary weight of $\mathbf{x}_i^s$ determined by the action of the DRL-based data selector.

The margin disparity discrepancy is a divergence measure with rigorous generalization bounds used to compare the distribution difference and ease the minimax optimization. In the adversarial learning setting, the margin disparity discrepancy [32] can be approximated as:

$$\begin{aligned} L_d = &\frac{\omega}{n_s} \sum_{i=1}^{n_s} \log[\sigma_{o(C(F(\mathbf{x}_i^s)))}(C'(F(\mathbf{x}_i^s)))] \\ &+ \frac{1}{n_t} \sum_{i=1}^{n_t} \log[1 - \sigma_{o(C(F(\mathbf{x}_i^t)))}(C'(F(\mathbf{x}_i^t)))], \end{aligned} \tag{12}$$

where $o()$ indicates the label obtained through one-hot encoding and ω is a weighting constant. Since margin disparity discrepancy is defined as the supremum over a hypothesis space, its minimization is a minimax optimization problem which can be implemented by a gradient reversal layer (GRL).

In addition, since it is revealed that features with smaller norms can be excessively less informative while larger norms are more transferable, we properly lift the concerned samples towards large-norm regions to promote successful knowledge transfer while eliminating domain shift. Hence, the parameters of the feature extractor F are updated by enlarging the feature norms of the target and concerned source samples. To achieve feature norm enlargement, the feature norm discrepancy L_s is formulated as:

$$\begin{aligned} L_s = &\frac{1}{n_s} \sum_{i=1}^{n_s} w(\mathbf{x}_i^s)[h(\mathbf{x}_i^s; \theta_0) + \Delta r - h(\mathbf{x}_i^s)]^2 \\ &+ \frac{1}{n_t} \sum_{j=1}^{n_t} [h(\mathbf{x}_j^t; \theta_0) + \Delta r - h(\mathbf{x}_j^t)]^2, \end{aligned} \tag{13}$$

where $h(x) = (\|\cdot\|_2 \circ F)(x)$, θ_0 represents the parameters of F updated from the last iteration, Δr is a positive residual scalar to control the norm enlargement so that the transferable features can be learned stably in a progressive manner.

Therefore, the optimization problem in the proposed domain adaptation model is stated as:

$$\begin{aligned} &\min_{F,C} L_y + L_d + \lambda L_s, \\ &\max_{C'} L_d, \end{aligned} \tag{14}$$

where λ is a hyperparameter to trade off the objectives. In this way, the domain shift is mitigated via margin disparity discrepancy minimization and feature norm enlargement so that the yielded classifier can achieve better generalization capability on the target domain.

IV. Experiments

Extensive experiments have been conducted on benchmark datasets to evaluate the performance of RAN and compare it with the state-of-the-art methods.

A. Setup

Office-31 [35] is a widely adopted domain adaptation dataset. It is relatively small and contains 4,652 images in 31 categories from three domains, i.e., Amazon (**A**), Webcam (**W**) and DSLR (**D**). Following the settings in [16], we select images from the same ten categories as target domains so that six PDA tasks can be created.

Office-Home [36] is a larger dataset which includes about 15,500 images in 65 categories from four different domains, i.e., Artistic (**Ar**), Clipart (**Cl**), Product (**Pr**), Real-World (**Rw**). Following the settings in [17], we select images from the first 25 categories in alphabetic order as target domains. Meanwhile, we use images from all the 65 categories as source domains to create twelve PDA tasks.

VisDA2017 [37] is a challenging large-scale dataset with over 280,000 images across 12 categories. It aims to bridge the significant gap between synthetic and real domains. Following the settings in [17], the first six categories in alphabetic order are chosen as target categories to create the Synthetic-12 $\rightarrow$ Real-6 task.

B. Implementation Details

The proposed RAN model is compared with 17 state-of-the-art methods. The experiments are implemented in Python on a desktop with one NVIDIA 1080 Ti GPU, Intel Core i7-7700 CPU of 3.6GHz and 32GB memory. In RAN, the feature extractor F consists of a backbone network, and an additional bottleneck layer. The classifier C includes two fully connected layers and the DRL Network contains three fully connected layers. During training, the backbone network is pre-trained on ImageNet while the new layers are trained from scratch. The domain adaptation model is trained using SGD with batch size 32 and learning rate 1e-3, and the DDQN model is trained using Adam with batch size 32 and learning rate 1e-4. The hyperparameters γ, ω, Δr and λ in Eq. 9, 12, 13 and 14 are set to 0.9, 4, 1 and 0.05, respectively. For all the baseline methods, we either refer to the reported results in [19], [20], [24]–[30] or calculate the average values of three runs using the original code. For fair comparison, RAN is also trained three times and the average values are calculated for evaluation.

TABLE I: Accuracy (%) on *Office-31* (ResNet-50)

Method	Office-31						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet [33]	75.59	96.27	98.09	83.44	83.92	84.97	87.05
DAN [7]	59.32	73.90	90.45	61.78	74.95	67.64	71.34
DANN [9]	73.56	96.27	98.73	81.53	82.78	86.12	86.50
ADDA [2]	75.67	95.38	99.85	83.41	83.62	84.25	87.03
RTN [22]	78.98	93.22	85.35	77.07	89.25	89.46	85.56
SAN [16]	93.90	99.32	99.36	94.27	94.15	88.73	94.96
IWAN [18]	89.15	99.32	99.36	90.45	95.62	94.26	94.69
PADA [17]	86.54	99.32	**100.00**	82.17	92.69	95.41	92.69
ETN [24]	94.52	**100.00**	**100.00**	95.03	**96.21**	94.64	96.73
SAFN [25]	87.34	97.74	99.36	90.02	92.69	93.32	93.41
DRCN [26]	90.80	**100.00**	**100.00**	94.30	95.20	94.80	95.90
DAPDA [27]	95.06	**100.00**	**100.00**	92.15	95.13	**97.40**	96.62
DCDF [28]	95.93	99.66	**100.00**	98.09	95.09	95.51	97.38
AGAN [29]	97.28	**100.00**	**100.00**	94.26	95.72	95.72	97.16
MSAN [30]	95.26	**100.00**	**100.00**	**100.00**	95.45	95.69	97.03
RTNet [19]	96.20	**100.00**	**100.00**	97.60	92.30	95.40	96.90
DARL [20]	94.58	99.66	**100.00**	98.73	94.57	94.26	96.97
RAN	**98.98±0.59**	**100.00±0**	**100.00±0**	97.67±0.97	96.28±0.34	96.24±0.11	**98.19**
RAN w/o L_s	96.27±1.48	**100.00±0**	**100.00±0**	95.33±0.97	95.79±0.24	95.54±0.26	97.16
RAN w/o L_d	96.95±1.88	**100.00±0**	**100.00±0**	97.24±0.73	95.72±0.10	95.37±0.12	97.55
RAN w/o Selector	91.30±0.20	99.77±0.20	**100.00±0**	94.27±1.10	94.85±0.21	95.27±0.22	95.91
RAN w/o DRL Selector	91.41±0.52	99.89±0.20	**100.00±0**	94.05±0.73	94.64±0.12	95.41±0.11	95.90

C. Comparison with State-of-the-Art Methods

The classification results based on ResNet on the six tasks of *Office-31* are shown in Table I. It can be observed that RAN significantly outperforms the baseline methods and achieves state-of-the-art accuracy. It can be noticed that DAN, DANN, ADDA and RTN all lead to worse performance compared to ResNet. This phenomenon demonstrates that domain adaptation methods can hardly deal with partial transfer problems because directly aligning two domains with different label spaces can lead to severe performance degradation. In contrast, the PDA methods perform better than ResNet on most tasks due to their weighting schemes which mitigate negative transfer caused by outlier data.

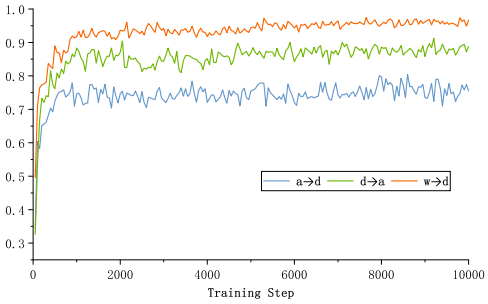
The classification results on the larger *Office-Home* dataset and the challenging large-scale *VisDa2017* dataset are shown in Table II and III, respectively. In these experiments, the domain adaptation methods achieve slightly better performance compared to ResNet. However, they still perform worse than most of the PDA algorithms, which demonstrates the significance of the elimination of irrelevant source samples. The superiority of RAN is even more noticeable when dealing with these more complex datasets. Generally, RAN continuously outperforms all the baseline methods on almost all tasks and improves the accuracy by a large margin. Moreover, it is noteworthy that the highest accuracy on each task is either achieved by our RAN model or the other two DRL-based models, RTNet and DARL. It implies that DRL algorithms can be applied to PDA tasks and are promising to promote knowledge transfer through filtering out irrelevant source data automatically.

In addition to classification results, the fractions of shared classes in the selected and eliminated samples are also analyzed respectively to verify the effectiveness of our DRL-based data selector. The results are depicted in Fig. 3. In line with our expectation, by adopting the proposed method, the fraction of shared classes in the selected samples becomes higher and higher as training proceeds. Meanwhile, the fraction of shared classes in the eliminated samples becomes lower and lower. This demonstrates that RAN has the ability to filter out irrelevant samples to promote knowledge transfer. Moreover, we also conduct sensitivity analysis of hyperparameters. The analysis on ω in Eq. 12 is shown in Fig. 4(a) and that on λ in Eq. 14 is shown in Fig. 4(b). According to the experimental results, it is found that although feature norm enlargement is promising to improve the performance, its effects should be controlled carefully within a reasonable range. Hence, the value of λ is set to its optimal value, i.e. 0.05, and kept constant in all the experiments. Differently, our model is not sensitive to hyperparameter ω and the change in accuracy caused by changing the value of ω is very small. In our experiments, we set the value of ω to 4.

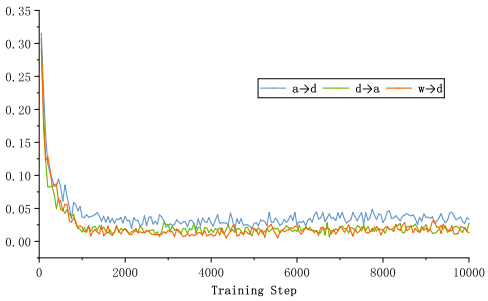
Generally, compared to the existing DRL-based PDA methods, RAN demonstrates outstanding superiority because: (1) Firstly, RAN enables the DRL and DA models to better leverage each other's strength. The reward function of RAN is meticulously designed so that the rewards are calculated directly based on the intermediate outputs of the DA model. Without introducing additional network blocks, such as the generators in RTNet, RAN directly links the two models together and strengthens their mutual assistance. Moreover, RAN also avoids the complexity caused by additional network blocks. (2) Secondly, RAN improves sample efficiency to better make use of every single experience. Different from RTNet which adopts an on-policy DRL algorithm, RAN employs an

TABLE II: Accuracy (%) on *Office-Home* (ResNet-50)

Method	Office-Home												
	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 [33]	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
DANN [9]	43.76	67.90	77.47	63.73	58.99	67.59	56.84	37.07	76.37	69.15	44.30	77.48	61.72
ADDA [2]	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
RTN [22]	49.31	57.70	80.07	63.54	63.47	73.38	65.11	41.73	75.32	63.18	43.57	80.50	63.07
SAN [16]	44.42	68.68	74.60	67.49	64.99	77.80	59.78	44.72	80.07	72.18	50.21	78.66	65.30
IWAN [18]	53.94	54.45	78.12	61.31	47.95	63.32	54.17	52.02	81.28	76.46	56.75	82.90	63.56
PADA [17]	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
ETN [24]	59.24	77.03	79.54	62.92	65.73	75.01	68.29	55.37	84.37	75.72	57.66	84.54	70.45
SAFN [25]	58.93	76.25	81.42	70.43	72.97	77.78	72.36	55.34	80.40	75.81	60.42	79.92	71.83
DRCN [26]	54.00	76.40	83.00	62.10	64.50	71.00	70.80	49.80	80.50	77.50	59.10	79.90	69.00
DAPDA [27]	56.49	77.56	80.29	65.73	71.52	77.28	66.53	55.96	85.65	77.02	60.82	84.82	71.64
DCDF [28]	60.30	80.17	81.23	67.49	68.24	76.04	68.31	55.05	83.77	75.39	58.93	83.14	71.51
AGAN [29]	56.36	77.25	85.09	74.20	73.84	81.12	70.80	51.52	84.54	78.97	56.78	83.42	72.82
MSAN [30]	59.28	77.59	82.50	64.00	68.24	75.48	68.87	51.10	83.27	76.78	59.82	82.80	70.80
RTNet [19]	63.20	80.10	80.70	66.70	69.30	77.20	71.60	53.90	84.60	77.40	57.90	85.50	72.30
DARL [20]	55.31	80.73	86.36	67.93	66.16	78.52	68.74	50.93	**87.74**	79.45	57.19	**85.60**	72.06
RAN	63.26	**83.08**	89.03	**74.99**	74.47	**82.90**	**77.99**	**61.19**	86.68	79.86	**63.52**	85.04	**76.84**
	±0.33	±0.69	±0.16	±0.70	±0.37	±0.28	±0.23	±0.36	±0.52	±0.14	±0.49	±0.20	
RAN w/o L_s	59.60	76.98	84.81	67.89	68.63	76.20	70.25	55.88	81.81	76.65	57.75	81.68	71.51
	±0.54	±0.16	±0.24	±0.74	±0.46	±0.78	±1.14	±0.25	±0.74	±0.06	±0.88	±0.88	
RAN w/o L_d	59.66	81.53	**89.18**	73.40	**74.90**	81.34	76.55	58.29	85.51	**80.99**	62.59	85.47	75.78
	±0.76	±0.18	±0.36	±0.23	±1.57	±0.44	±0.60	±1.36	±0.27	±0.65	±0.40	±0.35	
RAN w/o Selector	62.25	80.02	85.22	71.96	72.38	81.30	75.30	59.04	84.35	77.99	62.13	81.79	74.48
	±0.54	±0.48	±0.50	±0.88	±1.91	±0.95	±0.42	±0.49	±0.25	±0.66	±0.43	±0.30	
RAN w/o DRL Selector	**63.68**	81.42	87.28	73.95	72.87	81.87	77.87	60.54	85.42	78.79	63.13	83.40	75.85
	±0.25	±0.58	±0.55	±0.32	±1.85	±0.64	±0.79	±0.41	±0.31	±0.80	±0.78	±0.28	



(a) Fraction of Shared Classes in Selected Samples



(b) Fraction of Shared Classes in Eliminated Samples

Fig. 3: Fractions of shared classes in selected and eliminated samples with respect to number of iterations on the a → d, d → a and w → d tasks.

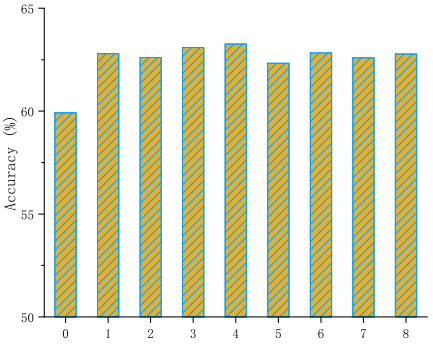
TABLE III: Accuracy (%) on *VisDa2017* (ResNet-50)

Method	Synthetic-12 → Real-6
ResNet [33]	45.26
DAN [7]	47.60
DANN [9]	51.01
RTN [22]	50.04
SAN [16]	49.90
IWAN [18]	48.60
PADA [17]	53.53
ETN [24]	69.20
SAFN [25]	67.65
DRCN [26]	58.20
AGAN [29]	67.71
DARL [20]	67.77
RAN	**75.10±2.90**

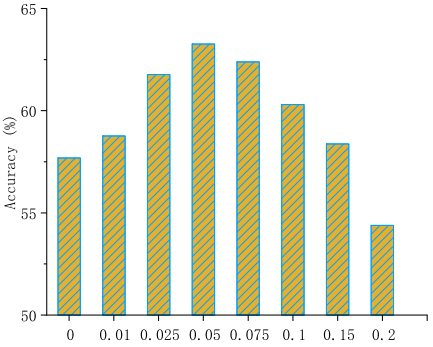
off-policy DRL algorithm. Since on-policy methods can only use data collected corresponding to the most recent policy while off-policy methods are able to reuse previously collected experiences performed by any policy for learning [38], RAN is more sample efficient than RTNet and is able to get more out of every sample through reusing the collected experiences. (3) Moreover, RAN generates more experiences to improve learning through enriched experiences. In DARL, since each state needs to consider the whole batch of source feature vectors while the selection is terminated whenever a negative reward is received, much less experiences can be generated.

TABLE IV: Evaluation of Generalization Capability of DRL-based Data Selector on *Office-31* (ResNet-50)

Method	Office-31						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg
SAFN [25]	87.34	97.74	99.36	90.02	92.69	93.32	93.41
SAFN w/ DRL-based Data Selector	**96.96**	**100.00**	**100.00**	**96.39**	**95.44**	**96.10**	**97.48**
AR [34]	93.54	100.00	**99.67**	96.82	95.51	96.04	96.93
AR w/ DRL-based Data Selector	**96.38**	**100.00**	99.36	**97.45**	**96.28**	**96.45**	**97.65**

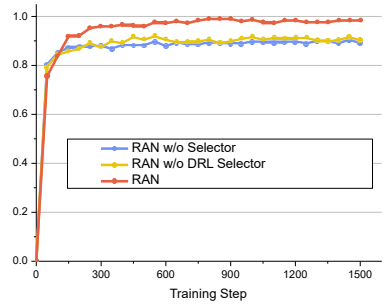


(a) ω

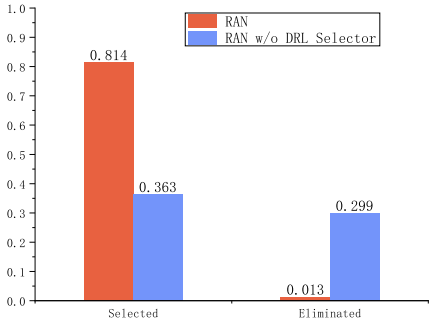


(b) λ

Fig. 4: Hyperparameter analysis conducted on the Ar → Cl task.



(a) Test Accuracy



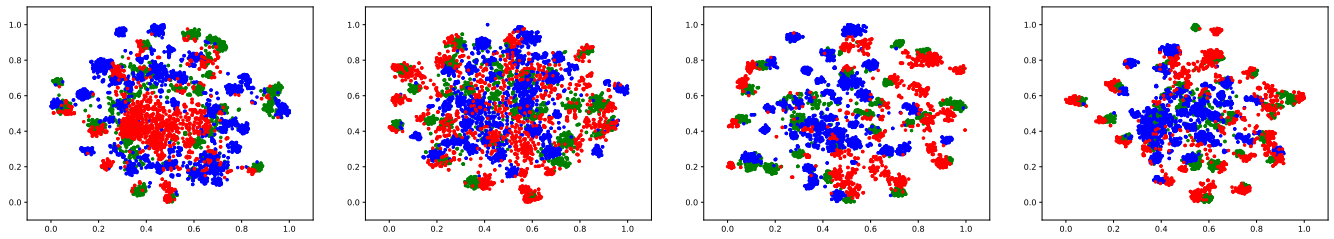
(b) Fraction of Shared Classes

Fig. 5: Test accuracy and fraction of shared classes on the a → w task.

In contrast, since each state in RAN is designed to consider only one source data instead of the whole batch of source data each time, a batch of n_b source samples can generate exactly n_b experiences. In this way, RAN can generate much more experiences for training. (4) Additionally, the DRL-based data selector in RAN is generic. Different from DARL which couples DRL with a specific DA method to calculate rewards, the proposed DRL-based data selector only relies on the feature extractor of the DA model, which is a general module in all DA models. Therefore, RAN is generic and can also be integrated into any DA method.

D. Ablation Study

To investigate the importance of different components of the proposed method, including the feature norm discrepancy, margin disparity discrepancy and DRL-based data selector, RAN is compared with four additional variants. The first (RAN w/o L_s) and second (RAN w/o L_d) variants are to verify the effectiveness of feature norm discrepancy and margin disparity discrepancy, respectively. In the third variant (RAN w/o Selector), the proposed domain adaptation model is trained using all the source instances without any data selector. In the fourth variant (RAN w/o DRL Selector), the selection of a source sample is determined directly based on its transferability instead of the action yielded by the DRL model. That is, a source instance is selected if its dissimilarity to the target domain is smaller than the batch average dissimilarity. Hence, the last two variants are to validate the superiority of the proposed DRL-based data selector. The comparison results are presented in Table I and II. Additionally, the test accuracy on the transfer task a → w during training is depicted in Fig. 5(a). The performance of RAN w/o L_s, RAN w/o L_d and RAN w/o Selector demonstrates the significance of the proposed DRL-based data selector since the largest performance degradation is caused by removing the DRL-based selector on the *Office-31* tasks while RAN w/o L_s even achieves satisfying results on the much more challenging *Office-Home* tasks solely through the DRL-based selector without any partial domain adaptation



(a) ResNet (b) $L_y + L_d$ (c) $L_y + L_s$ (d) $L_y + L_d + L_s$

Fig. 6: The t-SNE visualization of the Pr $\rightarrow$ Cl task with domain information. The blue dots represent the source feature points in the unshared 40 categories, the green dots represent the source feature points in the shared 25 categories, and the red dots represent the target feature points (25 categories).

loss. Meanwhile, the effectiveness of feature norm discrepancy and margin disparity discrepancy in the proposed mutually supportive framework is also illustrated by the results. Moreover, it is noticed that RAN w/o Selector and RAN w/o DRL Selector achieve similar accuracies on the *Office-31* tasks while both lead to much worse performance than RAN on both the *Office-31* and *Office-Home* tasks. These results reveal that a simple similarity-based data selector is not competent to promote knowledge transfer while the proposed DRL-based data selector is crucial.

Moreover, the fraction of shared classes in the selected and eliminated source data are shown in Fig. 5(b). It can be observed that most of the source data selected by RAN belong to the classes shared between the source and target domains, and most of the source data eliminated by RAN do not belong to the shared classes. In contrary, the simple similarity-based data selector does not demonstrate this property obviously. Therefore, it is proved that RAN is much more capable of selecting relevant source instances while eliminating irrelevant ones to achieve better knowledge transfer.

In addition, to evaluate the generalization capability of the proposed DRL-based source data selector, we directly integrate it with the SAFN algorithm [25] and the AR algorithm [34] without modifying either the DA algorithms or the DRL-based source data selector. As shown in Table IV, since our DRL-based data selector is generic, it is thereby capable of further improving the domain adaptation performance dramatically when combined with the DA algorithms. Besides, it is worth mentioning that during the training, we only calculate the classification and L2 norm losses based on the selected source samples while computing the adversarial loss based on all source samples. The purpose is to verify and guarantee the generalizability of the proposed DRL-based data selector because only the classification and L2 norm losses are generic and can be used in all DA algorithms. However, in fact, the proposed method can achieve even better performance when calculating all the three losses based merely on the selected source samples. According to our experiments, when all the three losses are calculated based only on the selected source samples, the RAN variant can achieve an average accuracy of 76.89 on the *Office-Home* tasks while outperforming the original RAN on 8 out of the 12 tasks.

The t-SNE [39] embeddings of the features are depicted in Fig. 6 to investigate the influence of the loss terms intuitively. The source feature points belonging to the categories which are shared between the source and target domains are represented in green while the source points in the unshared classes are shown in blue. The target feature points are represented in red. It can be observed that the target features extracted directly by the pre-trained backbone network, ResNet, are mixed together and messed up, indicating that the source and target correlated features are not well aligned. In contrast, RAN succeeds in discriminating different classes in both source and target domains and aligning the target samples to their corresponding source domain clusters. By adding the feature norm discrepancy L_s, RAN is able to pick up the most relevant source instances to train the adaptive classifier while eliminating the irrelevant ones to avoid negative transfer. As a result, its target feature points are less scattered, and the target and selected source data clusters become more noticeable. Meanwhile, the addition of the margin disparity discrepancy L_d enables better alignment of the two domains with improved cohesion in clusters and thus results in more compact features.

E. Future Work

Despite the superiority of RAN, it still has a lot of room for improvement. Some interesting future work worthy of exploration are as follows. First, the current transferability of source samples are calculated within one batch while the sampled target data may not well represent the whole target domain data. As a result, batch size can be a hyperparameter which limits the adaptation performance. For instance, RAN achieves an average accuracy of 96.86, 97.14, 98.2 on the *Office-31* tasks with batch size 8, 16 and 32, respectively. Hence, in the future, the entire target feature can be encoded and thereby used to measure the transferability more accurately and efficiently. In addition, the reward function can also be redesigned to include more feedback signals, such as the diversity of selected source samples, to further boost knowledge transfer. Moreover, the proposed DRL-based data selector is also promising to be extended to deal with more challenging scenarios, such as universal domain adaptation [40], [41].

V. Conclusion

In this paper, we propose a novel DRL-based PDA method named Reinforced Adaptation Network (RAN) to promote cross-domain knowledge transfer in partial domain adaptation tasks. In RAN, a DRL model and a domain adaptation model are jointly trained. The DRL-based data selector is developed for the automatic elimination of irrelevant source instances to circumvent negative transfer. Meanwhile, the domain adaptation model is to mitigate domain shift via margin disparity discrepancy minimization and progressive feature norm enlargement. Experimental results have demonstrated that RAN significantly outperforms the state-of-the-art partial domain adaptation methods.

Acknowledgement

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151) and Career Development Award (Grant No. C210112046).

References

[1] W. Deng, L. Zheng, Y. Sun, and J. Jiao, "Rethinking triplet loss for domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 29–37, 2020.

[2] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[3] L. Zhang, P. Wang, W. Wei, H. Lu, C. Shen, A. van den Hengel, and Y. Zhang, "Unsupervised domain adaptation using robust class-wise matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1339–1349, 2018.

[4] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[5] Z. Chen, C. Chen, X. Jin, Y. Liu, and Z. Cheng, "Deep joint two-stream wasserstein auto-encoder and selective attention alignment for unsupervised domain adaptation," *Neural Computing and Applications*, pp. 1–14, 2019.

[6] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2208–2217.

[7] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *arXiv preprint arXiv:1502.02791*, 2015.

[8] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[9] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[10] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[11] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.

[12] A. Chadha and Y. Andreopoulos, "Improving adversarial discriminative domain adaptation," *arXiv preprint arXiv:1809.03625*, 2018.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[14] X. Xu, H. He, H. Zhang, Y. Xu, and S. He, "Unsupervised domain adaptation via importance sampling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4688–4699, 2019.

[15] Y. Tian and S. Zhu, "Partial domain adaptation on semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[16] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2724–2732.

[17] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–150.

[18] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8156–8164.

[19] Z. Chen, C. Chen, Z. Cheng, B. Jiang, K. Fang, and X. Jin, "Selective transfer with reinforced transfer network for partial domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 706–12 714.

[20] J. Chen, X. Wu, L. Duan, and S. Gao, "Domain adversarial reinforcement learning for partial domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

[21] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[22] 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.

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

[24] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2985–2994.

[25] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1426–1435.

[26] S. Li, C. H. Liu, Q. Lin, Q. Wen, L. Su, G. Huang, and Z. Ding, "Deep residual correction network for partial domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[27] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3404–3416, 2021.

[28] T. Jing, M. Shao, and Z. Ding, "Discriminative cross-domain feature learning for partial domain adaptation," *arXiv preprint arXiv:2008.11360*, 2020.

[29] Y. Kim and S. Hong, "Adaptive graph adversarial networks for partial domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[30] C. Zhang and Q. Zhao, "Attention guided for partial domain adaptation," *Information Sciences*, vol. 547, pp. 860–869, 2021.

[31] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[32] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," *arXiv preprint arXiv:1904.05801*, 2019.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[34] X. Gu, X. Yu, J. Sun, Z. Xu *et al.*, "Adversarial reweighting for partial domain adaptation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 860–14 872, 2021.

[35] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[36] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

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

[38] L. Graesser and W. L. Keng, *Foundations of deep reinforcement learning: theory and practice in Python*. Addison-Wesley Professional, 2019.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[40] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2720–2729.

[41] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," *Advances in neural information processing systems*, vol. 33, pp. 16 282–16 292, 2020.