Learning to Transfer Examples for Partial Domain Adaptation

Zhangjie Cao[1,*], Kaichao You[1,*], Mingsheng Long[1]([✉]), Jianmin Wang[1], and Qiang Yang[2]

[1]KLiss, MOE; BNRist; School of Software, Tsinghua University, China

[1]Research Center for Big Data, Tsinghua University, China

[1]Beijing Key Laboratory for Industrial Big Data System and Application

[2]Hong Kong University of Science and Technology, China

{caozhangjie14, youkaichao}@gmail.com, {mingsheng, jimwang}@tsinghua.edu.cn

Abstract

Domain adaptation is critical for learning in new and unseen environments. With domain adversarial training, deep networks can learn disentangled and transferable features that effectively diminish the dataset shift between the source and target domains for knowledge transfer. In the era of Big Data, large-scale labeled datasets are readily available, stimulating the interest in partial domain adaptation (PDA), which transfers a recognizer from a large labeled domain to a small unlabeled domain. It extends standard domain adaptation to the scenario where target labels are only a subset of source labels. Under the condition that target labels are unknown, the key challenges of PDA are how to transfer relevant examples in the shared classes to promote positive transfer and how to ignore irrelevant ones in the source domain to mitigate negative transfer. In this work, we propose a unified approach to PDA, Example Transfer Network (ETN), which jointly learns domain-invariant representations across domains and a progressive weighting scheme to quantify the transferability of source examples. A thorough evaluation on several benchmark datasets shows that ETN consistently achieves state-of-the-art results for various partial domain adaptation tasks.

1. Introduction

Deep neural networks have significantly advanced the state-of-the-art performance for various machine learning problems [13, 15] and applications [11, 20, 30]. A common prerequisite of deep neural networks is the rich labeled data to train a high-capacity model to have sufficient generalization power. Such rich supervision is often prohibitive in real-world applications due to the huge cost of data annotation. Thus, to reduce the labeling cost, there is a strong need to develop versatile algorithms that can leverage rich labeled

---

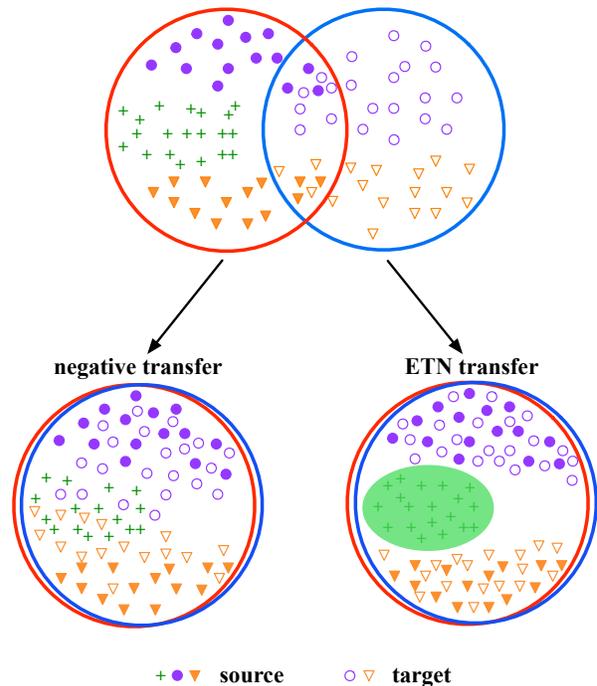[*]Equal contribution, in alphabetic order

Figure 1. Partial domain adaptation (PDA) is a generalized setting of domain adaptation where the source label space subsumes the target label space. The technical challenge of PDA lies in an intrinsic negative transfer caused by the outlier source classes ('+' in this case), which cannot be forcefully transferred to the target domain. The proposed Example Transfer Network (ETN) designs a weighting scheme to quantify the transferability of source examples and only transfer source examples relevant to the target domain (purple circle and orange triangle), eliminating outlier source examples (in green shadow). Source and target domains are denoted by red and blue circles respectively. *Best viewed in color.*

data from a related source domain. However, this domain adaptation paradigm is hindered by the dataset shift underlying different domains, which forms a major bottleneck to adapting the category models to novel target tasks [29, 36].

A major line of the existing domain adaptation methods bridge different domains by learning domain-invariant feature representations in the absence of target labels, *i.e.*, unsupervised domain adaptation. Existing methods assume that the source and target domains share the same set of class labels [32, 12], which is crucial for directly applying the source-trained classifier to the target domain. Recent studies in deep learning reveal that deep networks can disentangle explanatory factors of variations behind domains [8, 42], thus learning more transferable features to improve domain adaptation significantly. These deep domain adaptation methods typically embed distribution matching modules, including moment matching [38, 21, 22, 23] and adversarial training [10, 39, 37, 24, 17], into deep architectures for end-to-end learning of transferable representations.

Although existing methods can reduce the feature-level domain shift, they assume label spaces across domains are identical. In real-world applications, it is often formidable to find a relevant dataset with the label space identical to the target dataset of interest which is often unlabeled. A more practical scenario is Partial Domain Adaptation (PDA) [5, 43, 6], which assumes that the source label space is a superspace of the target label space, relaxing the constraint of identical label spaces. PDA enables knowledge transfer from a big domain of many labels to a small domain of few labels. With the emergence of Big Data, large-scale labeled datasets such as ImageNet-1K [31] and Google Open Images [19] are readily accessible to empower data-driven artificial intelligence. These repositories are almost universal to subsume categories of the target domain, making PDA feasible to many applications. PDA can also work in the regime where target data are in limited categories. For example, functions of protein are limited. A large database of known protein structures can be collected, which includes all functions. For a new species, proteins have different structures, but their functions are contained in the database. Predicting protein functions for new species falls into the PDA problem.

As a generalization to standard domain adaptation, partial domain adaptation is more challenging: the target labels are unknown at training, and there must be many "outlier" source classes that are useless for the target task. This technical challenge is intuitively illustrated in Figure 1, where the target classes (like purple '○' and orange '▽') will be forcefully aligned to the outlier source classes (like '+') by existing domain adaptation methods. As a result, negative transfer will happen because the learner migrates harmful knowledge from the source domain to the target domain. Negative transfer is the principal obstacle to the application of domain adaptation techniques [29].

Thus, matching the whole source and target domains as previous methods [21, 10] is not a safe solution to the PDA problem. We need to develop algorithms versatile enough to transfer useful examples from the many-class dataset (source domain) to the few-class dataset (target domain) while robust enough to irrelevant or outlier examples. Three approaches to partial domain adaptation [5, 43, 6] address the PDA by weighing each data point in the domain-adversarial networks, where a domain discriminator is learned to distinguish the source and target. While decreasing the impact of irrelevant examples on domain alignment, they do not undo the negative effect of the outlier classes on the source classifier. Moreover, they evaluate the transferability of source samples without considering the underlying discriminative and multimodal structures. As a result, it is still vulnerable that they may align the features of outlier source classes and target classes, giving way to negative transfer.

Towards a safe approach to partial domain adaptation, we present the Example Transfer Network (ETN), which improves the previous work [5, 43, 6] by learning to transfer useful examples. ETN automatically evaluates the transferability of source examples with a transferability quantifier based on their similarities to the target domain, which is used to weigh their contributions to both the source classifier and the domain discriminator. In particular, ETN improves the weight quality over previous work [43] by further revealing the discriminative structure to the transferability quantifier. By this means, irrelevant source examples can be better detected and filtered out. Another key improvement of ETN over the previous methods is the capability to simultaneously confine the source classifier and the domain-adversarial network within the auto-discovered shared label space, thus promoting the positive transfer of relevant examples and mitigating negative transfer of irrelevant examples. Comprehensive experiments demonstrate that our model achieves state-of-the-art results on several benchmark datasets, including Office-31, Office-Home, ImageNet-1K, and Caltech-256.

2. Related Work

Domain Adaptation Domain adaptation, a special scenario of transfer learning [29], bridges domains of different distributions to mitigate the burden of annotating target data for machine learning [28, 9, 44, 41], computer vision [32, 12, 16] and natural language processing [7]. The main technical difficulty of domain adaptation is to formally reduce the distribution discrepancy across different domains. Deep networks can learn representations that suppress explanatory factors of variations behind data [3] and manifest invariant factors across different populations. These invariant factors enable knowledge transfer across relevant domains [42]. Deep networks have been extensively explored for domain adaptation [27, 16], yielding significant performance gains against shallow domain adaptation methods.

While deep representations can disentangle complex data distributions, recent advances show that they can only reduce, but not remove, the cross-domain discrepancy [38]. Thus deep learning alone cannot bound the generalization risk for

the target task [25, 1]. Recent works bridge deep learning and domain adaptation [38, 21, 10, 39, 22]. They extend deep networks to domain adaptation by adding adaptation layers through which high-order statistics of distributions are explicitly matched [38, 21, 22], or by adding a domain discriminator to distinguish features of the source and target domains, while the features are learned adversarially to deceive the discriminator in a minimax game [10, 39].

Partial Domain Adaptation While the standard domain adaptation advances rapidly, it still needs the vanilla assumption that the source and target domains share the same label space. This assumption does not hold in partial domain adaptation (PDA), which transfers models from many-class domains to few-class domains. There are three valuable efforts towards the PDA problem. Selective Adversarial Network (SAN) [5] adopts multiple adversarial networks with a weighting mechanism to select out source examples in the outlier classes. Partial Adversarial Domain Adaptation [6] improves SAN by employing only one adversarial network and further adding the class-level weight to the source classifier. Importance Weighted Adversarial Nets (IWAN) [43] uses the Sigmoid output of an auxiliary domain classifier (not involved in domain-adversarial training) to derive the probability of a source example belonging to the target domain, which is used to weigh source examples in the domain-adversarial network. These pioneering approaches achieve dramatical performance gains over standard methods in partial domain adaptation tasks.

These efforts mitigate negative transfer caused by outlier source classes and promote positive transfer among shared classes. However, as outlier classes are only selected out for the domain discriminators, the source classifier is still trained with all classes [5], whose performance for shared classes may be distracted by outlier classes. Further, the domain discriminator of IWAN [43] for obtaining the importance weights distinguishes the source and target domains only based on the feature representations, without exploiting the discriminative information in the source domain. This will result in non-discriminative importance weights to distinguish shared classes from outlier classes. This paper proposes an Example Transfer Network (ETN) that down-weights the irrelevant examples of outlier classes further on the source classifier and adopts a discriminative domain discriminator to quantify the example transferability.

Open-Set Domain Adaptation On par with domain adaptation, research has been dedicated to open set recognition, with the goal to reject outliers while correctly recognizing inliers during testing. Open Set SVM [18] trains a probabilistic SVM and rejects unknown samples by a threshold. Open Set Neural Network [2] generalizes deep neural networks to open set recognition by introducing an OpenMax layer, which estimates the probability of an input from an unknown class and rejects the unknown point by a threshold.

Open Set Domain Adaptation (OSDA) [4, 33] tackles the setting when the training and testing data are from different distributions and label spaces. OSDA methods often assume which classes are shared by the source and target domains are known at training. Unlike OSDA, in our scenario, target classes are entirely unknown at training. It is interesting to extend our work to the open set scenario under the generic assumption that all target classes are unknown.

3. Example Transfer Network

The scenario of partial domain adaptation (PDA) [5] constitutes a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples associated with $|\mathcal{C}_s|$ classes and a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples drawn from $|\mathcal{C}_t|$ classes. Note that in PDA the source domain label space $\mathcal{C}_s$ is a superspace of the target domain label space $\mathcal{C}_t$ i.e. $\mathcal{C}_s \supset \mathcal{C}_t$. The source and target domains are drawn from different probability distributions p and q respectively. Besides $p \neq q$ as in standard domain adaptation, we further have $p_{\mathcal{C}_t} \neq q$ in partial domain adaptation, where $p_{\mathcal{C}_t}$ denotes the distribution of the source domain data in label space $\mathcal{C}_t$. The goal of PDA is to learn a deep network that enables end-to-end training of a transferable feature extractor G_f and an adaptive classifier G_y to sufficiently close the distribution discrepancy across domains and bound the target risk $\Pr_{(\mathbf{x},y) \sim q} [G_y(G_f(\mathbf{x})) \neq \mathbf{y}]$.

We incur deteriorated performance when directly applying the source classifier G_y trained with standard domain adaptation methods to the target domain. In partial domain adaptation, it is difficult to identify which part of the source label space $\mathcal{C}_s$ is shared with the target label space $\mathcal{C}_t$ because the target domain is fully unlabeled and $\mathcal{C}_t$ is unknown at the training stage. Under this condition, most of existing deep domain adaptation methods [21, 10, 39, 22] are prone to *negative transfer*, a degenerated case where the classifier with adaptation performs even worse than the classifier without adaptation. The negative transfer happens since they assume that the source and target domains have identical label space and match whole distributions p and q even though $p_{\mathcal{C}_s \setminus \mathcal{C}_t}$ and q are non-overlapping and cannot be matched in principle. Thus, decreasing the negative effect of the source examples in outlier label space $\mathcal{C}_s \setminus \mathcal{C}_t$ is the key to mitigating negative transfer in partial domain adaptation. Besides, we also need to reduce the distribution shift across $p_{\mathcal{C}_t}$ and q to enhance positive transfer in the shared label space $\mathcal{C}_t$ as before. Note that the irrelevant source examples may come from both outlier classes and shared classes, thus requiring a versatile algorithm to identify them.

3.1. Transferability Weighting Framework

The key technical problem of domain adaptation is to reduce the distribution shift between the source and target domains. Domain adversarial networks [10, 39] tackle this
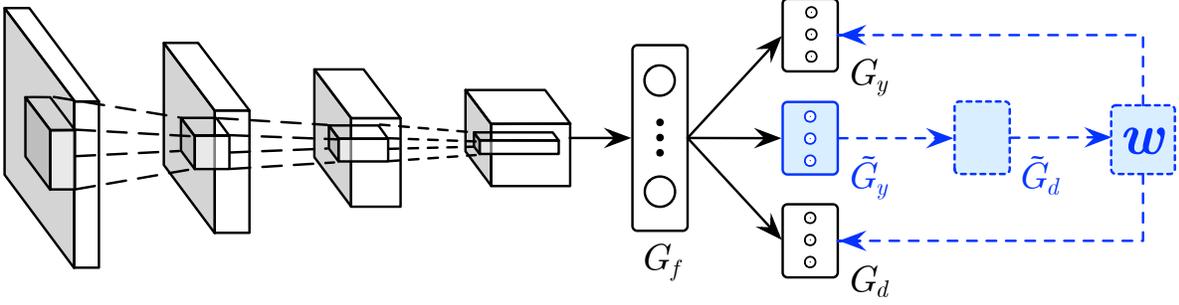
Figure 2. Example Transfer Network (ETN) for partial domain adaptation, where G_f is the feature extractor, G_y is the source classifier, G_d is domain discriminator (involved in adversarial training) for domain alignment; $\tilde{G}_d$ is the auxiliary domain discriminator (uninvolved in adversarial training) that quantifies the transferability $\mathbf{w}$ of each source example, and $\tilde{G}_y$ is the auxiliary label predictor encoding the discriminative information to the auxiliary domain discriminator $\tilde{G}_d$. Modules in blue are newly designed in this paper. *Best viewed in color.*

problem by learning transferable features in a two-player minimax game: the first player is a domain discriminator G_d trained to distinguish the feature representations of the source domain from the target domain, and the second player is a feature extractor G_f trained simultaneously to deceive the domain discriminator.

Specifically, the domain-invariant features $\mathbf{f}$ are learned in a minimax optimization procedure: the parameters θ_f of the feature extractor G_f are trained by maximizing the loss of domain discriminator G_d, while the parameters θ_d of the domain discriminator G_d are trained by minimizing the loss of the domain discriminator G_d. Note that our goal is to learn a source classifier that transfers to the target, hence the loss of the source classifier G_y is also minimized. This leads to the optimization problem proposed in [10]:

$$
\begin{aligned}
E\left(\theta_f, \theta_y, \theta_d\right) = & \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), \mathbf{y}_i\right) \\
& - \frac{1}{n_a} \sum_{\mathbf{x}_i \in \mathcal{D}_a} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), \mathbf{d}_i\right),
\end{aligned}
\tag{1}
$$

where $\mathcal{D}_a = \mathcal{D}_s \cup \mathcal{D}_t$ is the union of the source and target domains and $n_a = |\mathcal{D}_a|$, $\mathbf{d}_i$ is the domain label, L_y and L_d are the cross-entropy loss functions.

While domain adversarial networks yield reliable results for standard domain adaptation, they will incur performance degeneration on the partial domain adaptation where $\mathcal{C}_s \supset \mathcal{C}_t$. This degeneration is caused by the outlier classes $\mathcal{C}_s \backslash \mathcal{C}_t$ in the source domain, which are undesirably matched to the target classes $\mathcal{C}_t$. Due to the domain gap, even the source examples in the shared label space $\mathcal{D}_t$ may not transfer well to the target domain. As a consequence, we need to design a new framework for partial domain adaptation.

This paper presents a new transferability weighting framework to address the technical difficulties of partial domain adaptation. Denote by $w(\mathbf{x}_i^s)$ the weight of each source example $\mathbf{x}_i^s$, which quantifies the example's transferability. Then for a source example with a larger weight, we should

increase its contribution to the final model to enhance positive transfer; otherwise, we should decrease its contribution to mitigating negative transfer. IWAN [43], a previous work for partial domain adaptation, reweighs the source examples in the loss of the domain discriminator G_d. We further put the weights in the loss of the source classifier G_y. This significantly enhances our ability to diminish the irrelevant source examples that deteriorate our final model.

Furthermore, the unknownness of target labels can make the identification of shared classes difficult, making partial domain adaptation more difficult. We thus believe that the exploitation of unlabeled target examples by semi-supervised learning is also indispensable. We make use of the entropy minimization principle [14]. Let $\hat{\mathbf{y}}_j^t = G_y(G_f(\mathbf{x}_j^t)) \in \mathbb{R}^{|\mathcal{C}_s|}$, the entropy loss to quantify the uncertainty of a target example's predicted label is $H\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right) = -\sum_{c=1}^{|\mathcal{C}_s|} \hat{y}_{j,c}^t \log \hat{y}_{j,c}^t$.

The transferability weighting framework is shown in Figure 2. By weighting the losses of the source classifier G_y and the domain discriminator G_d using the transferability $w(\mathbf{x}_i^s)$ of each source example, and combining the entropy minimization criterion, we achieve the following objective:

$$
\begin{aligned}
E_{G_y} = & \frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) L\left(G_y\left(G_f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right)\right) \\
& + \frac{\gamma}{n_t} \sum_{j=1}^{n_t} H\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
E_{G_d} = & -\frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) \log\left(G_d\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) \\
& - \frac{1}{n_t} \sum_{j=1}^{n_t} \log\left(1 - G_d\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),
\end{aligned}
\tag{3}
$$

where γ is a hyper-parameter to trade-off the labeled source examples and unlabeled target examples.

The transferability weighting framework can be trained end-to-end by a minimax optimization procedure as follows,

yielding a saddle point solution $(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d)$:

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E_{G_y} - E_{G_d},$$
$$(\hat{\theta}_d) = \arg\min_{\theta_d} E_{G_d}. \qquad (4)$$

3.2. Example Transferability Quantification

With the proposed transferability weighting framework in Equations (2) and (3), the key technical problem is how to quantify the transferability of each source example $w(\mathbf{x}_i^s)$. We introduce an auxiliary domain discriminator $\tilde{G}_d$, which is also trained to distinguish the representations of the source domain from the target domain, using the similar loss as Equation (3) but dropping $w(\mathbf{x}_i^s)$. It is not involved in the adversarial training procedure, *i.e.*, the features G_f are not learned to confuse $\tilde{G}_d$. Such an auxiliary domain discriminator can roughly quantify the transferability of the source examples, through the Sigmoid probability of classifying each source example $\mathbf{x}_i^s$ to the target domain.

Such an auxiliary domain discriminator $\tilde{G}_d$ discriminates source and target domains based on the assumption that source examples of shared classes $\mathcal{C}_t$ are closer to the target domain than to those source examples in the outlier classes $\mathcal{C}_s \backslash \mathcal{C}_t$, thus having higher probability to be predicted as from the target domain. However, the auxiliary domain discriminator only distinguishes the source and target examples based on domain information. There is potential small gap between $\tilde{G}_d$'s outputs for transferable and irrelevant source examples especially when $\tilde{G}_d$ is trained well. So the model is still exposed to the risk of mixing up the transferable and irrelevant source examples, yielding unsatisfactory transferability measures $w(\mathbf{x}_i^s)$. In partial domain adaptation, the source examples in $\mathcal{C}_t$ differentiate from those in $\mathcal{C}_s \backslash \mathcal{C}_t$ mainly in that $\mathcal{C}_t$ is shared with the target domain while $\mathcal{C}_s \backslash \mathcal{C}_t$ has no overlap with the target domain. Thus, it is natural to integrate discriminative information into our weight design to resolve the ambiguity between shared and outlier classes.

Inspired by AC-GANs [26] that integrate the labeled information into the discriminator, we aim to integrate the label information into the auxiliary domain discriminator $\tilde{G}_d$. However, we hope to develop a transferability measure $w(\mathbf{x}_i^s)$ with both the discriminative information and domain information to generate clearly separable weights for source data in $\mathcal{C}_t$ and $\mathcal{C}_s \backslash \mathcal{C}_t$ respectively. Thus, we add an auxiliary label predictor $\tilde{G}_y$ with leaky-softmax activation. Within $\tilde{G}_y$, the feature from feature extractor G_f are transformed to $|\mathcal{C}_s|$-dimension $\mathbf{z}$. Then $\mathbf{z}$ will be passed through a leaky-softmax activation as follows,

$$\tilde{\sigma}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{|\mathcal{C}_s| + \sum_{c=1}^{|\mathcal{C}_s|} \exp(z_c)}, \qquad (5)$$

where z_c is the c-th dimension of $\mathbf{z}$. The leaky-softmax has the property that the element-sum of its outputs is smaller

than 1; when the logit z_c of class c is very large, the probability to classify an example as class c is high. As the auxiliary label predictor $\tilde{G}_y$ is trained on source examples and labels, the source examples will have higher probability to be classified as a specific source class c, while the target examples will have smaller logits and uncertain predictions. Therefore, the element-sum of the leaky-softmax outputs are closer to 1 for source examples and closer to 0 for target examples. If we define $\tilde{G}_d$ as

$$\tilde{G}_d(G_f(\mathbf{x}_i)) = \sum_{c=1}^{|\mathcal{C}_s|} \tilde{G}_y^c(G_f(\mathbf{x}_i)), \qquad (6)$$

where $\tilde{G}_y^c(G_f(\mathbf{x}_i))$ is the probability of each example $\mathbf{x}_i$ belonging to class c, then $\tilde{G}_d(G_f(\mathbf{x}_i))$ can be seen as computing the probability of each example belonging to the source domain. For a source example, the smaller the value of $\tilde{G}_d(G_f(\mathbf{x}_i))$ is, the more probable that it comes from the target domain, meaning that it is closer to the target domain and more likely to be in the shared label space $\mathcal{C}_t$. Thus, the output of $\tilde{G}_d$ is suitable for transferability quantification.

We train the auxiliary label predictor $\tilde{G}_y$ with the leaky-softmax by a multitask loss over $|\mathcal{C}_s|$ one-vs-rest binary classification tasks for the $|\mathcal{C}_s|$-class classification problem:

$$E_{\tilde{G}_y} = -\frac{\lambda}{n_s} \sum_{i=1}^{n_s} \sum_{c=1}^{|\mathcal{C}_s|} \Big[y_{i,c}^s \log \tilde{G}_y^c(G_f(\mathbf{x}_i^s)) + (1 - y_{i,c}^s) \log \left(1 - \tilde{G}_y^c(G_f(\mathbf{x}_i^s))\right) \Big], \qquad (7)$$

where $y_{i,c}^s$ denotes whether class c is the ground-truth label for source example $\mathbf{x}_i^s$, and λ is a hyper-parameter. We also train the auxiliary domain discriminator $\tilde{G}_d$ to distinguish the features of the source domain and the target domain as

$$E_{\tilde{G}_d} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log \left(\tilde{G}_d(G_f(\mathbf{x}_i^s)) \right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log \left(1 - \tilde{G}_d(G_f(\mathbf{x}_j^t)) \right). \qquad (8)$$

From Equations (6) to (8), we observe that the outputs of the auxiliary domain discriminator $\tilde{G}_d$ depend on the outputs of the auxiliary label predictor $\tilde{G}_y$. This guarantees that $\tilde{G}_d$ is trained with both label and domain information, resolving the ambiguity between shared and outlier classes to better quantify the example transferability.

Finally, with the help of the auxiliary label predictor $\tilde{G}_y$ and the auxiliary domain discriminator $\tilde{G}_d$, we can derive more accurate and discriminative weights to quantify the transferability of each source example as

$$w(\mathbf{x}_i^s) = 1 - \tilde{G}_d(G_f(\mathbf{x}_i^s)). \qquad (9)$$

Since the outputs of $\tilde{G}_d$ for source examples are closer to 1, implying very small weights, we normalize the weights in each mini-batch of batch size B as $w(\mathbf{x}) \leftarrow \frac{w(\mathbf{x})}{\frac{1}{B} \sum_{i=1}^{B} w(\mathbf{x}_i)}$.

3.3. Minimax Optimization Problem

With the aforementioned derivation, we now formulate our final model, Example Transfer Network (ETN). We unify the transferability weighting framework in Equations (2)–(3) and the example transferability quantification in Equations (6)–(9). Denoting by $\theta_{\tilde{y}}$ the parameters of the auxiliary label predictor $\tilde{G}_y$, the proposed ETN model can be solved by a minimax optimization problem that finds saddle-point solutions $\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d$ and $\hat{\theta}_{\tilde{y}}$ to model parameters as follows,

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f, \theta_y}{\arg\min}\ E_{G_y} - E_{G_d},$$
$$(\hat{\theta}_d) = \underset{\theta_d}{\arg\min}\ E_{G_d}, \qquad (10)$$
$$(\hat{\theta}_{\tilde{y}}) = \underset{\theta_{\tilde{y}}}{\arg\min}\ E_{\tilde{G}_y} + E_{\tilde{G}_d}.$$

ETN enhances partial domain adaptation by learning to transfer relevant examples and diminish outlier examples for both source classifier G_y and domain discriminator G_d. It exploits progressive weighting schemes $w(\mathbf{x}_i^s)$ from the auxiliary domain discriminator $\tilde{G}_d$ and auxiliary label predictor $\tilde{G}_y$, well quantifying the transferability of source examples.

4. Experiments

We conduct experiments to evaluate our approach with state-of-the-art (partial) domain adaptation methods. Codes and datasets will be available at github.com/thuml.

4.1. Setup

Office-31 [32] is *de facto* for domain adaptation. It is relatively small with 4,652 images in 31 classes. Three domains, namely **A, D, W**, are collected by downloading from **amazon.com** (**A**), taking from **DSLR** (**D**) and from **web camera** (**W**). Following the protocol in [5], we select images from the 10 categories shared by **Office-31** and **Caltech-256** to build new target domain, creating six partial domain adaptation tasks: **A→W, D→W, W→D, A→D, D→A** and **W→A**. Note that there are 31 categories in the source domain and 10 categories in the target domain.
Office-Home [40] is a larger dataset, with 4 domains of distinct styles: **Artistic**, **Clip Art**, **Product** and **Real-World**. Each domain contains images of 65 object categories. Denoting them as **Ar, Cl, Pr, Rw**, we obtain twelve partial domain adaptation tasks: **Ar→Cl, Ar→Pr, Ar→Rw, Cl→Ar, Cl→Pr, Cl→Rw, Pr→Ar, Pr→Cl, Pr→Rw, Rw→Ar, Rw→Cl**, and **Rw→Pr**. For PDA, we use images from the first 25 classes in alphabetical order as the target domain and images from all 65 classes as the source domain.
ImageNet-Caltech is a large dataset built with **ImageNet-1K** [31] and **Caltech-256**. They share 84 classes, and thus we form two partial domain adaptation tasks: **ImageNet (1000)→Caltech (84)** and **Caltech (256)→ImageNet (84)**.

As most networks are trained on the training set of ImageNet, we use images from ImageNet validation set as target domain for **Caltech (256)→ImageNet (84)** task.

We compare the proposed **ETN** model with state-of-the-art deep learning and (partial) domain adaptation methods: **ResNet-50** [15], Deep Adaptation Network (**DAN**) [21], Domain-Adversarial Neural Networks (**DANN**) [10], Adversarial Discriminative Domain Adaptation (**ADDA**) [37], Residual Transfer Networks (**RTN**) [22], Selective Adversarial Network (**SAN**) [5], Importance Weighted Adversarial Network (**IWAN**) [43] and Partial Adversarial Domain Adaptation (**PADA**) [6].

Besides ResNet-50 [15], we also evaluate ETN and some methods based on **VGG** [34] on the Office-31 dataset. We perform ablation study to justify the example transfer mechanism, by evaluating two **ETN** variants: 1) **ETN w/o classifier** is the variant without weights on the source classifier; 2) **ETN w/o auxiliary** is the variant without the auxiliary label predictor on the auxiliary domain discriminator.

We implement all methods based on **PyTorch**, and fine-tune ResNet-50 [15] and VGG [34] pre-trained on ImageNet. New layers are trained from scratch, and their learning rates are 10 times that of the fine-tuned layers. We use mini-batch SGD with momentum of 0.9 and the learning rate decay strategy implemented in DANN [10]: the learning rate is adjusted during SGD using $\eta_p = \frac{\eta_0}{(1+\alpha p)^\beta}$, where p is the training progress linearly changing from 0 to 1. The flip-coefficient of the gradient reversal layer is increased gradually from 0 to 1 as DANN [10]. Hyper-parameters are optimized with importance weighted cross-validation [35].

4.2. Results

The classification results based on **ResNet-50** on the the twelve tasks of **Office-Home**, six tasks of **Office-31** and the two large-scale tasks of **ImageNet-Caltech** are shown in Tables 1 and 2. We also compare all methods on **Office-31** with **VGG** backbone in Table 3. ETN outperforms all other methods w.r.t average accuracy, showing that ETN performs well with different base networks on different datasets.

Specifically, we have several observations. **1)** ADDA, DANN, and DAN outperform ResNet only on some tasks, implying that they suffer from the negative transfer issue. **2)** RTN exploits the entropy minimization criterion to amend itself with semi-supervised learning. Thus, it has some improvement over ResNet but still suffers from negative transfer for some tasks. **3)** Partial domain adaptation methods (SAN [5] and IWAN [43]) perform better than ResNet and other domain adaptation methods on most tasks, due to their weighting mechanism to mitigate negative transfer caused by outlier classes and promote positive transfer among shared classes. **4)** ETN outperforms SAN and IWAN on most tasks, showing its power to discriminate the outlier classes from the shared classes accurately and to transfer relevant examples.

Table 1. Classification Accuracy (%) for Partial Domain Adaptation on Office-Home Dataset (**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 [15]	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 [10]	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 [37]	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
IWAN [43]	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
SAN [5]	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
PADA [6]	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	**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**

Table 2. Classification Accuracy (%) for Partial Domain Adaptation on Office-31 and ImageNet-Caltech Datasets (**ResNet-50**)

Method	Office-31							ImageNet-Caltech		Avg
	A→W	D→W	W→D	A→D	D→A	W→A	Avg	I → C	C → I	
ResNet [15]	75.59±1.09	96.27±0.85	98.09±0.74	83.44±1.12	83.92±0.95	84.97±0.86	87.05±0.94	69.69±0.78	71.29±0.74	70.49±0.76
DAN [21]	59.32±0.49	73.90±0.38	90.45±0.36	61.78±0.56	74.95±0.67	67.64±0.29	71.34±0.46	71.30±0.46	60.13±0.50	65.72±0.48
DANN [10]	73.56±0.15	96.27±0.26	98.73±0.20	81.53±0.23	82.78±0.18	86.12±0.15	86.50±0.20	70.80±0.66	67.71±0.76	69.23±0.71
ADDA [37]	75.67± 0.17	95.38±0.23	99.85±0.12	83.41± 0.17	83.62±0.14	84.25±0.13	87.03±0.16	71.82±0.45	69.32±0.41	70.57±0.43
RTN [22]	78.98±0.55	93.22±0.52	85.35±0.47	77.07±0.49	89.25±0.39	89.46±0.37	85.56±0.47	75.50±0.29	66.21±0.31	70.85±0.30
IWAN [43]	89.15±0.37	99.32±0.32	99.36±0.24	90.45±0.36	95.62±0.29	94.26±0.25	94.69±0.31	78.06±0.40	73.33±0.46	75.70±0.43
SAN [5]	93.90±0.45	99.32±0.52	99.36±0.12	94.27±0.28	94.15±0.36	88.73±0.44	94.96±0.36	77.75±0.36	**75.26**±0.42	76.51±0.39
PADA [6]	86.54±0.31	99.32±0.45	**100.00**±.00	82.17±0.37	92.69±0.29	**95.41**±0.33	92.69±0.29	75.03±0.36	70.48±0.44	72.76±0.40
ETN	**94.52**±0.20	**100.00**±.00	**100.00**±.00	**95.03**±0.22	**96.21**±0.27	94.64±0.24	**96.73**±0.16	**83.23**±0.24	74.93±0.28	**79.08**±0.26

Table 3. Classification Accuracy (%) for Partial Domain Adaptation on Office-31 Dataset (**VGG**)

Method	Office-31						
	A→W	D→W	W→D	A→D	D→A	W→A	Avg
VGG [34]	60.34±0.84	97.97±0.63	99.36±0.36	76.43±0.48	72.96±0.56	79.12±0.54	81.03± 0.57
DAN [21]	58.78±0.43	85.86±0.32	92.78±0.28	54.76±0.44	55.42±0.56	67.29±0.20	69.15±0.37
DANN [10]	50.85±0.12	95.23±0.24	94.27±0.16	57.96±0.20	51.77±0.14	62.32±0.12	68.73±0.16
ADDA [37]	53.28±0.15	94.33±0.18	95.36±0.08	58.78±0.12	50.24±0.10	63.34±0.08	69.22±0.12
RTN [22]	69.35±0.42	98.42±0.48	99.59±0.32	75.43±0.38	81.45±0.32	82.98±0.36	84.54±0.38
IWAN [43]	82.90±0.31	79.75±0.26	88.53±0.16	**90.95**±0.33	89.57±0.24	93.36±0.22	87.51±0.25
SAN [5]	83.39±0.36	99.32±0.45	**100.00**±.00	90.70±0.20	87.16±0.23	91.85±0.35	92.07±0.27
PADA [6]	**86.05**±0.36	99.42±0.24	**100.00**±.00	81.73±0.34	93.00±0.24	**95.26**±0.27	92.54±0.24
ETN	85.66±0.16	**100.00**±.00	**100.00**±.00	89.43±0.17	**95.93**±0.23	92.28±0.20	**96.74**±0.13

Table 4. Classification Accuracy (%) of ETN and Its Variants for Partial Domain Adaptation on Office-Home Dataset (**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
ETN w/o classifier	56.18	71.93	79.32	**65.11**	65.57	73.66	65.47	52.90	82.88	72.93	56.93	82.91	68.93
ETN w/o auxiliary	48.36	50.42	79.13	56.57	45.88	65.49	56.38	49.07	77.53	75.57	**58.81**	78.32	61.79
ETN	**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**

In particular, ETN outperforms SAN and IWAN by much larger margin on the large-scale **ImageNet-Caltech** dataset, indicating that ETN is robuster to outlier classes and performs better even on dataset with large number of outlier classes (916 in **ImageNet→Caltech**) relative to the shared classes (84 in **ImageNet→Caltech**). ETN has two advantages: learning discriminative weights and filtering outlier classes out from both source classifier and domain discriminator, which boost partial domain adaptation performance.

We inspect the efficacy of different modules by comparing in Tables 4 the results of ETN variants. **1)** ETN outperforms ETN w/o classifier, proving that the weighting mechanism

on the source classifier can reduce the negative influence of outlier-classes examples and focus the source classifier on the examples belonging to the target label space. **2)** ETN also outperforms ETN w/o auxiliary by a larger margin, proving that the auxiliary classifier can inject label information into the domain discriminator to yield discriminative weights, which in turn enables ETN to filter out irrelevant examples.

4.3. Analysis

Feature Visualization: We plot in Figures 3 the t-SNE embeddings [8] of the features learned by DANN, SAN, IWAN and ETN on **A (31 classes)→W (10 classes)** with class in-

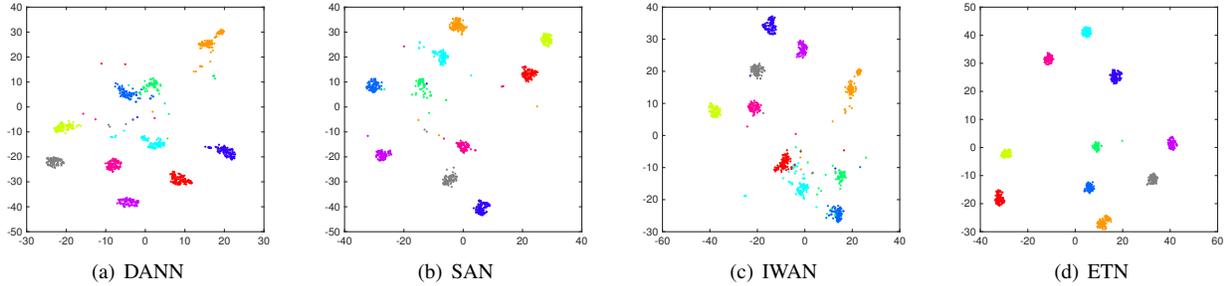| (a) DANN | (b) SAN | (c) IWAN | (d) ETN |

Figure 3. Visualization of features learned by DANN, SAN, IWAN, and ETN (class information is denoted by different colors).

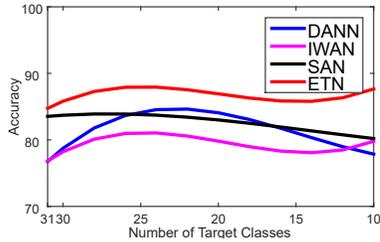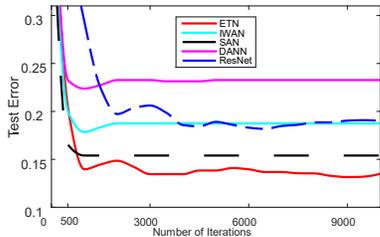Figure 4. Accuracy by varying #target classes.

Figure 5. Target test error w.r.t. to #iterations.
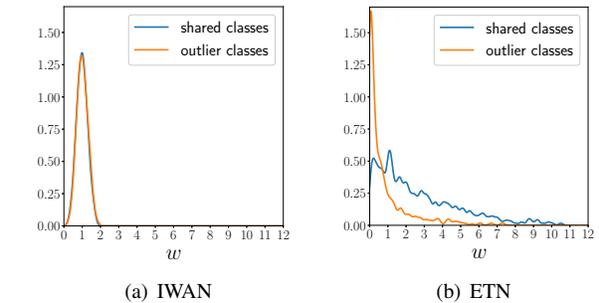

| (a) IWAN | (b) ETN |

Figure 6. Density function of the importance weights of source examples in the shared label space $\mathcal{C}_t$ and outlier label space $\mathcal{C}_s \backslash \mathcal{C}_t$.

formation in the target domain. We observe that features learned by DANN, IWAN, and SAN are not clustered as clearly as ETN, indicating that ETN can better discriminate target examples than the compared methods.

Class Overlap: We conduct a wide range of partial domain adaptation with different numbers of target classes. Figure 4 shows that when the number of target classes decreases fewer than 23, the performance of DANN degrades quickly, implying that negative transfer becomes severer when the label space overlap becomes smaller. The performance of SAN decreases slowly and stably, indicating that SAN potentially eliminates the influence of outlier classes. IWAN only performs better than DANN when the label space non-overlap is very large and negative transfer is very severe. ETN performs stably and consistently better than all compared methods, showing the advantage of ETN to partial domain adaptation. ETN also performs better than DANN in standard domain adaptation when the label spaces totally overlap, implying that the weighting mechanism will not degrade performance when there are no outlier classes.

Convergence Performance: As shown in Figure 5, the test errors of all methods converge fast but baselines to high error

rates. Only ETN converges to the lowest test error. Such phenomenon implies that ETN can be trained more efficiently and stably than previous domain adaptation methods.

Weight Visualization: We plot the approximate density function of the weights in Equation (9) generated by IWAN [43] and ETN for all source examples in Figure 6 on task **Cl→Pr**. The orange curve shows examples in shared classes $\mathcal{C}_t$ and the blue curve shows outlier classes $\mathcal{C}_s \backslash \mathcal{C}_t$. Compared to IWAN, our ETN approach assigns much larger weights to shared classes and much smaller weights to outlier classes. Most examples of outlier classes have nearly zero weights, explaining the strong performance of ETN on these datasets.

5. Conclusion

This paper presented Example Transfer Network (ETN), a discriminative and robust approach to partial domain adaptation. It quantifies the transferability of source examples by integrating the discriminative information into the transferability quantifier and down-weights the negative influence of the outlier source examples upon both the source classifier and the domain discriminator. Based on the evaluation, our model performs strongly for partial domain adaptation tasks.

Acknowledgements

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[2] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572, 2016.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, 2013.

[4] P. P. Busto and J. Gall. Open set domain adaptation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 754–763, 2017.

[5] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Partial transfer learning with selective adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.

[9] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(3):465–479, 2012.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35, 2016.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

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

[14] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 529–536. MIT Press, 2005.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1994–2003, 2018.

[18] Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision (ECCV)*, pages 393–409, 2014.

[19] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.

[22] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 136–144, 2016.

[23] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2208–2217, 2017.

[24] Zelun Luo, Yuliang Zou, Judy Hoffman, and Fei-Fei Li. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 164–176, 2017.

[25] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Computational Learning Theory (COLT)*, 2009.

[26] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2642–2651, 2017.

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

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

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

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99. 2015.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

[33] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision (ECCV)*, September 2018.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR), 2015 (arXiv:1409.1556v6)*, 2015.

[35] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Muller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 8(May):985–1005, 2007.

[36] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011.

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. 2014.

[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Simultaneous deep transfer across domains and tasks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] X. Wang and J. Schneider. Flexible transfer learning under support and model shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[42] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[43] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[44] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning (ICML)*, 2013.