Universal Domain Adaptation

Kaichao You[1], Mingsheng Long[1](✉), Zhangjie Cao[1], Jianmin Wang[1], and Michael I. Jordan[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]University of California, Berkeley, USA
youkaichao@gmail.com, {mingsheng,jimwang}@tsinghua.edu.cn, jordan@cs.berkeley.edu

Abstract

Domain adaptation aims to transfer knowledge in the presence of the domain gap. Existing domain adaptation methods rely on rich prior knowledge about the relationship between the label sets of source and target domains, which greatly limits their application in the wild. This paper introduces Universal Domain Adaptation (UDA) that requires no prior knowledge on the label sets. For a given source label set and a target label set, they may contain a common label set and hold a private label set respectively, bringing up an additional category gap. UDA requires a model to either (1) classify the target sample correctly if it is associated with a label in the common label set, or (2) mark it as "unknown" otherwise. More importantly, a UDA model should work stably against a wide spectrum of commonness (the proportion of the common label set over the complete label set) so that it can handle real-world problems with unknown target label sets. To solve the universal domain adaptation problem, we propose Universal Adaptation Network (UAN). It quantifies sample-level transferability to discover the common label set and the label sets private to each domain, thereby promoting the adaptation in the automatically discovered common label set and recognizing the "unknown" samples successfully. A thorough evaluation shows that UAN outperforms the state of the art closed set, partial and open set domain adaptation methods in the novel UDA setting.

1. Introduction

Deep learning has boosted the progress of computer vision and improved state of the art performance on diverse vision tasks such as image classification [13], object detection [30] and semantic segmentation [12]. However, the remarkable efficacy of deep learning algorithms highly relies on abundant labeled training data, which requires tedious labor work on collecting labeled data. Given a large-scale unlabeled dataset, it is usually prohibitive to annotate enough training data such that we can train a deep learning model that generalizes well. An alternative is to leverage off-the-shelf labeled data from a related domain (source domain) to improve the model for the domain of interest (target domain). The target domain may contain data collected by different sensors, from different perspectives or under different illumination conditions compared with the source domain, leading to large **domain gap**. Domain adaptation [33] aims to minimize the domain gap and successfully transfer the model trained on the source domain to the target domain.

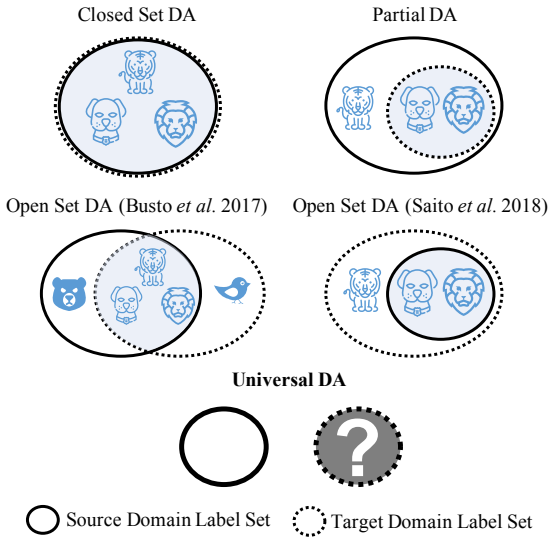


Figure 1. Universal Domain Adaptation (UDA) and existing domain adaptation settings with respect to label sets of source and target domains (blue shades indicate shared labels). Only UDA is able to deal with the setting that the label set of target domain is unknown.

Existing domain adaptation methods tackle the domain gap either by learning domain invariant feature representation, by generating features/samples for target domains or by transforming samples between domains through generative

models. They suppose that label sets are identical across domains, as shown in Figure 1 (closed set domain adaptation). This simplified scenario focuses on the fundamental problem of domain adaptation and provides insightful ideas for future research. Recent works try to relax the assumption by proposing open set domain adaptation [28, 35] and partial domain adaptation [2, 45]. As shown in Figure 1, partial domain adaptation [2, 45] requests that the source label set contains the target label set while Busto *et al.* [28] introduces "unknown" classes in both domains, and assumes common classes between two domains are known in the training phase. Modified open set domain adaptation by Saito *et al.* [35] removes data of source unknown classes such that the source label set is a subset of the target label set. Luo *et al.* [24] allows partly shared label sets and requires some labeled data in the target domain, where the target label set is known. These works constitute valuable advances towards practical domain adaptation.

Practical scenarios are way more complicated and these assumptions are easily violated. For example, labeled animals from different datasets are easily accessible. But if we want to recognize animals in the wild, we are exposed to two challenges: **(1)** The background may deviate from those in the training data, leading to large **domain gap**; **(2)** Some native species do not exist in the training data, in the meantime, animal species in the deployed environment may not cover all the training species because training data is too diverse, leading to large **category gap**. In summary, the relationship of label sets between the source and target domains is unknown in the presence of a large domain gap. If the source label set is large enough to contain the target label set, partial domain adaptation methods are good choices; if the source label set is contained in the target label set or common classes are known, open set domain adaptation methods are good choices. In a general scenario, however, we cannot select the proper domain adaptation method because no prior knowledge about the target domain label set is given.

For this purpose, we propose a generalized setting, termed **Universal Domain Adaptation** (**UDA**). In UDA, given a labeled source domain, for any related target domain, regardless of how its label set differs from that of the source domain, we need to classify its samples correctly if it belongs to any class in the source label set, or mark it as"unknown" otherwise. The word *"universal"* indicates that UDA imposes no prior knowledge on the label sets.

UDA poses two major technical challenges for designing domain adaptation models in the wild. (**1**) Since we know nothing about the target label set, we cannot decide which part of the source domain should be matched to which part of the target domain. If we naively match the entire source domain with the entire target domain, mismatching of different label sets will deteriorate the model. (**2**) The model should be able to mark target samples as "unknown" if they do not belong to any class in the source label set. Since there are no labeled training data for these classes, by no means the classifier can tell their detailed category.

To address Universal Domain Adaptation, we propose **Universal Adaptation Network** (**UAN**), equipping with a novel criterion to quantify the transferability of each sample. The criterion integrates both the domain **similarity** and the prediction **uncertainty** of each sample into a sample-level weighting mechanism. With the transferability-enhanced UAN model, the samples coming from the common label set between the source and target domains are automatically detected and matched while the target samples coming from the target private label set can be successfully marked by a rejection pipeline as "unknown" class.

The main contributions of this paper are:

(1) We introduce a more practical Universal Domain Adaptation (UDA) setting that imposes no prior knowledge on the label sets of source and target domains. This is important considering that we do not have access to target labels in unsupervised domain adaptation and sometimes it is even impossible to know the target label set, not to mention how it overlaps with the source label set.

(2) We study the performance of existing domain adaptation methods under a variety of UDA settings including closed set, partial and open set domain adaptation. Methods tailored to specific settings do not work well in UDA. This highlights the need for a UDA-friendly model.

(3) We propose Universal Adaptation Network (UAN), an end-to-end solution, which exploits both the domain similarity and the prediction uncertainty of each sample to develop a weighting mechanism for discovering label sets shared by both domains and promote common-class adaptation. Empirical results show that UAN works stably across different UDA settings and outperforms existing methods.

2. Related Work

We briefly review recent domain adaptation methods in this section. According to the constraint on the label set relationship between domains, these methods fall into closed set domain adaptation, partial domain adaptation, or open set domain adaptation.

2.1. Closed Set Domain Adaptation

Closed set domain adaptation focuses on mitigating the impact of the domain gap between source and target domains. Solutions to closed set domain adaptation mainly fall into two categories: feature adaptation and generative model. Feature adaptation methods diminish the feature distribution discrepancy between source and target domains by minimizing well-defined statistical distances on feature distributions. Early shallow adaptation methods [33, 7, 27, 5, 46, 42] usually provide insights in developing modern deep adaptation methods [38, 21, 6, 11, 39, 23, 37, 34, 22], while other

deep adaptation methods further explore architecture designs [19, 43, 36, 26, 20, 41, 16, 47, 25, 4, 18]. Tzeng *et al.* [38] and Long *et al.* [21] first proposed to minimize Maximum Mean Discrepancy (MMD) of deep features across domains. Long *et al.* [23] further exploits a residual transfer structure and introduces entropy minimization on target data. Zellinger *et al.* [44] enables distribution alignment by optimizing Central Moment Discrepancy (CMD). Haeusser *et al.* [11] constructs a bipartite graph to force feature distribution alignment within clusters. Bhushan *et al.* [1] enables domain adaptation by minimizing Earth Mover's Distance (EMD) between distributions. Meanwhile, with significant advances made in image synthesis by Generative Adversarial Nets [8], methods that match feature distributions by generative models are proposed. They learn a domain classifier to discriminate features from source and target domains and force the feature extractor to confuse the domain classifier in an adversarial learning paradigm [6, 39, 37].

Methods based on generative models synthesize labeled target samples as data augmentation and match domains in both pixel and feature levels [19, 36, 16, 20, 26, 17, 41]. With the impressive results of Cycle-Consistent Generative Adversarial Network [48] in image translation, CycleGAN-based domain adaptation methods have been studied recently [15, 32]. These methods usually transform source images into target-like images and vice versa with CycleGAN, then train the classifiers for each domain respectively with source images and transformed images.

Attempts for closed set domain adaptation focus on solving fundamental problems in distribution matching and provide a solid basis for the extension of domain adaptation.

2.2. Partial Domain Adaptation

The presence of Big Data gives rise to partial domain adaptation (PDA) [2, 45, 3], which transfers a learner from a big source domain to a small target domain. The label set of the source domain is supposed to be large enough to contain the target label set. To solve partial domain adaptation, Cao *et al.* [2] utilizes multiple domain discriminators with class-level and instance-level weighting mechanism to achieve per-class adversarial distribution matching. Zhang *et al.* [45] constructs an auxiliary domain discriminator to quantify the probability of a source sample being similar to the target domain. Cao *et al.* [3] further improves PDA by employing only one adversarial network and jointly applying class-level weighting on the source classifier.

Efforts for partial domain adaptation push well-studied domain adaptation problem towards a more practical setting.

2.3. Open Set Domain Adaptation

Busto *et al.* [28] proposed open set domain adaptation (OSDA), as shown in Figure 1. The classes private to both domains are unified as an "unknown" class. They use an Assign-and-Transform-Iteratively (ATI) algorithm to map target samples to source classes and then train SVMs for final classification. Saito *et al.* [35] modified the open set domain adaptation by requiring no data of the source private label set and extends the source classifier by adding an explicit "unknown" class and trains it adversarially among classes.

These methods tackle the domain gap by discarding the "unknown" classes when common classes are known in advance. While confined from more generalized settings, they shed light on designing practical domain adaptation models.

3. Universal Domain Adaptation

In this section, we formally introduce Universal Domain Adaptation (UDA) setting and address it by a novel Universal Adaptation Network (UAN).

3.1. Problem Setting

In Universal Domain Adaptation (UDA), a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}$ consisting of n_s labeled samples and a target domain $\mathcal{D}_t = \{(\mathbf{x}_i^t)\}$ of n_t unlabeled samples are provided at training. Note that the source data are sampled from distribution p while the target data from distribution q. We use $\mathcal{C}_s$ to denote the label set of source domain and $\mathcal{C}_t$ the label set of target domain. $\mathcal{C} = \mathcal{C}_s \cap \mathcal{C}_t$ is the common label set shared by both domains. $\overline{\mathcal{C}}_s = \mathcal{C}_s \setminus \mathcal{C}$ and $\overline{\mathcal{C}}_t = \mathcal{C}_t \setminus \mathcal{C}$ represent the label sets private to the source domain and the target domain respectively. $p_{\mathcal{C}_s}$ and $p_{\mathcal{C}}$ are used to denote the distributions of source data with labels in the label set $\mathcal{C}_s$ and $\mathcal{C}$ respectively, and $q_{\mathcal{C}_t}$, $q_{\mathcal{C}}$ for target distributions with labels in the label set $\mathcal{C}_t$, $\mathcal{C}$ respectively. Note that the target data are *fully unlabeled*, and the target label sets (inaccessible at training) are only used for defining the UDA problem.

We define the **commonness** between two domains as the Jaccard distance of two label sets: $\xi = \frac{|\mathcal{C}_s \cap \mathcal{C}_t|}{|\mathcal{C}_s \cup \mathcal{C}_t|}$. Closed set domain adaptation is a special case of UDA when $\xi = 1$. The smaller ξ is, the less sharing knowledge is and the more difficult the adaptation is. **The task for UDA is to design a model that does not know ξ but works well across a wide spectrum of ξ.** It must be able to distinguish between target data coming from $\mathcal{C}$ and target data coming from $\overline{\mathcal{C}}_t$, as well as to learn a classification model f to minimize the target risk in the common label set, i.e. $\min \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim q_{\mathcal{C}}} [f(\mathbf{x}) \neq \mathbf{y}]$.

3.2. Technical Challenges

In UDA, a new challenge has emerged, the **category gap** between two domains. The root of the category gap lies in the difference of the label sets. If we naively pick any of the existing closed set domain adaptation methods to solve UDA, source data in $\overline{\mathcal{C}}_s$ may be matched with target data from $\overline{\mathcal{C}}_t$. Such blind alignment is problematic since their label sets have no overlap ($\overline{\mathcal{C}}_s \cap \overline{\mathcal{C}}_t = \emptyset$) and forcefully matching them will cause many target private data to be

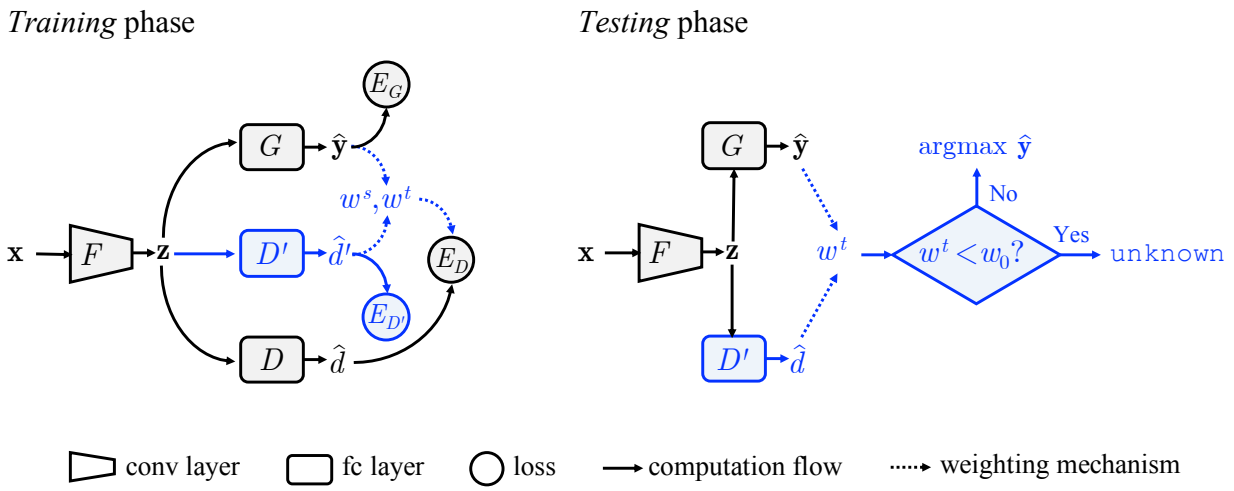


Figure 2. The training and testing phases of the Universal Adaptation Network (UAN) designed for Universal Domain Adaptation (UDA).

predicted as a class in $\overline{\mathcal{C}}_s$ whereas they should be marked as "unknown". If we turn to tailored methods of partial or open set domain adaptation, we must face the fact that the relationship between $\mathcal{C}_s$ and $\mathcal{C}_t$ is unknown. In the absence of the configuration about $\mathcal{C}$, $\overline{\mathcal{C}}_s$ and $\overline{\mathcal{C}}_t$, it is hard to make a choice among tailored domain adaptation methods. Thus, we need to automatically identify the source and target data from $\mathcal{C}$, such that feature alignment can be done in the auto-discovered common label set.

Despite the category gap, the **domain gap** still exists in UDA setting, i.e. between the source and target data in the common label set. In other words, $p \neq q$ and $p_{\mathcal{C}} \neq q_{\mathcal{C}}$. Domain adaptation should be applied to align distributions of the source and target data in the common label set $\mathcal{C}$.

Another challenge for UDA is to **detect "unknown" classes**. In practice, confidence thresholding, which marks samples with low classification confidence as "unknown", is often used. Nonetheless, such a straightforward method may fail in universal domain adaptation since the predictions by neural networks are usually overconfident [10] but less discriminative due to the underlying domain gap.

3.3. Universal Adaptation Network

We propose Universal Adaptation Network (UAN) to address the UDA problem. As shown in Figure 2, the architecture of UAN consists of a feature extractor F, an adversarial domain discriminator D, a non-adversarial domain discriminator D' and a label classifier G. Input $\mathbf{x}$ from either domain is fed into the feature extractor F. The extracted feature $\mathbf{z} = F(\mathbf{x})$ is forwarded into the label classifier G to obtain the probability $\hat{\mathbf{y}} = G(\mathbf{z})$ of $\mathbf{x}$ over the source classes $\mathcal{C}_s$. The non-adversarial domain discriminator D' obtains the **domain similarity** $\hat{d}' = D'(\mathbf{z})$, quantifying the similarity of $\mathbf{x}$ to the source domain. The adversarial domain discriminator D aims to adversarially match the feature distributions of the source and target data falling in the common label set $\mathcal{C}$ (Note that we need a mechanism to detect the common label set). E_G, $E_{D'}$ and E_D represent the error for label classifier G, non-adversarial domain discriminator D' and adversarial domain discriminator D, which are formally defined as

$$E_G = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p} L\left(\mathbf{y}, G(F(\mathbf{x}))\right) \tag{1}$$

$$\begin{aligned} E_{D'} = &- \mathbb{E}_{\mathbf{x}\sim p} \log D'\left(F\left(\mathbf{x}\right)\right) \\ &- \mathbb{E}_{\mathbf{x}\sim q} \log\left(1 - D'\left(F\left(\mathbf{x}\right)\right)\right) \end{aligned} \tag{2}$$

$$\begin{aligned} E_D = &- \mathbb{E}_{\mathbf{x}\sim p} w^s(\mathbf{x}) \log D\left(F\left(\mathbf{x}\right)\right) \\ &- \mathbb{E}_{\mathbf{x}\sim q} w^t(\mathbf{x}) \log\left(1 - D\left(F\left(\mathbf{x}\right)\right)\right) \end{aligned} \tag{3}$$

where L is the standard cross-entropy loss, $w^s(\mathbf{x})$ indicates the probability of a source sample $\mathbf{x}$ belonging to the common label set $\mathcal{C}$, and similarly, $w^t(\mathbf{x})$ indicates the probability of a target sample $\mathbf{x}$ belonging to the common label set $\mathcal{C}$. The details of $w^s(\mathbf{x})$ and $w^t(\mathbf{x})$ will be elaborated in the next subsection. With well-established weighting $w^s(\mathbf{x})$ and $w^t(\mathbf{x})$, the adversarial domain discriminator D is confined to distinguish the source and target data in the common label set $\mathcal{C}$. Adversarially, the feature extractor F strives to confuse D, yielding domain-invariant features in the common label set $\mathcal{C}$. The label classifier G trained on such features can be applied safely to the target domain.

The training of UAN can be written as a minimax game:

$$\begin{aligned} &\max_{D} \min_{F,G} E_G - \lambda E_D \\ &\min_{D'} E_{D'} \end{aligned} \tag{4}$$

where λ is a hyper-parameter to trade off between transferability and discriminability. We utilize the well-established gradient reversal layer proposed by Ganin *et al.* [6] to reverse

the gradient between F and D to optimize all the modules in an end-to-end training framework.

The testing phase of UAN is shown in the right plot of Figure 2. Given each input target sample $\mathbf{x}$, its categorical prediction $\hat{\mathbf{y}}(\mathbf{x})$ over the source label set $\mathcal{C}_s$, and the domain prediction $\hat{d}'(\mathbf{x})$, we compute $w^t(\mathbf{x})$ using Eq. (8) (details in the next subsection). With a validated threshold w_0, the class $y(\mathbf{x})$ can be predicted by thresholding $\hat{\mathbf{y}}(\mathbf{x})$ w.r.t. w_0:

$$y(\mathbf{x}) = \begin{cases} \texttt{unknown} & w^t < w_0 \\ \operatorname{argmax}(\hat{\mathbf{y}}) & w^t \geq w_0 \end{cases} \tag{5}$$

which either rejects the target sample $\mathbf{x}$ as "unknown" class or classifies it to one of the source classes.

3.4. Transferability Criterion

In this section, we further elaborate on how to compute weighting $w^s = w^s(\mathbf{x})$ and $w^t = w^t(\mathbf{x})$ by sample-level transferability criterion. With a proper sample-level transferability criterion, each point in both source and target domains can be weighted such that the distributions of source and target data in the common label set $\mathcal{C}$ can be maximally aligned. Also, data from target private label set $\overline{\mathcal{C}}_t$ can be identified and marked as "unknown" with the help of the sample-level transferability criterion. Thus, a well-established sample-level transferability criterion should satisfy Eq. (6):

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} w^s(\mathbf{x}) > \mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} w^s(\mathbf{x}) \\ &\mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} w^t(\mathbf{x}) > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} w^t(\mathbf{x}) \end{aligned} \tag{6}$$

These inequalities should hold in a non-negligible margin.

Now we need to construct the sample-level transferability criterion. We first list what we have at hand about each input $\mathbf{x}$: $\hat{y}$, $\hat{d}$, $\hat{d}'$. Since D is involved in adversarial training and thus fooled, its output $\hat{d}$ is not discriminative enough. We thus analyze the properties of $\hat{y}$ and $\hat{d}'$ as follows.

Domain Similarity. In Eq. (2), the objective of D' is to predict samples from source domain as 1 and samples from target domain as 0. Thus, $\hat{d}'$ can be seen as the quantification for the domain similarity of each sample. For a source sample, smaller $\hat{d}'$ means that it is more similar to the target domain; for a target sample, larger $\hat{d}'$ means that it is more similar to the source domain. Therefore, we can hypothesize that $\mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} \hat{d}'$.

Due to the nature of D', inequality $\mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} \hat{d}', \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} \hat{d}', \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} \hat{d}'$ naturally holds. Since $p_{\mathcal{C}}$ and $q_{\mathcal{C}}$ share the same label set, $p_{\mathcal{C}}$ is closer to $q_{\mathcal{C}}$ compared with $q_{\overline{\mathcal{C}}_t}$, and it is reasonable to hypothesize $\mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} \hat{d}'$. The same observation applies to $\mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} \hat{d}'$.

Prediction Uncertainty. The prediction $\hat{\mathbf{y}}$ contains the discriminative information about the input, but it is only reliable in the source domain guaranteed by labeled data. To exploit unlabeled data, entropy minimization has been used as a criterion in semi-supervised learning and domain adaptation [9, 23] to enforce the decision boundary in the unlabeled data to pass through low-density area. In principle, entropy quantifies the prediction uncertainty, and smaller entropy means more confident prediction. We hypothesize: $\mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} H(\hat{\mathbf{y}})$.

Since the source domain is labeled and the target domain is unlabeled, predictions are certain for source samples and uncertain for target samples, $\mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} H(\hat{\mathbf{y}}), \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} H(\hat{\mathbf{y}}), \mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} H(\hat{\mathbf{y}})$.

Similar samples from $q_{\mathcal{C}}$ and $p_{\mathcal{C}}$ can attract each other. Thus, the entropy of samples from $p_{\mathcal{C}}$ becomes larger because they are influenced by the high entropy samples from $q_{\mathcal{C}}$. Still, as $\overline{\mathcal{C}}_s$ has no intersection with $\mathcal{C}_t$, samples from $p_{\overline{\mathcal{C}}_s}$ are not influenced by the target data and keeps highest certainty. So we hypothesize that $\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} H(\hat{\mathbf{y}})$. Similarly, $\overline{\mathcal{C}}_t$ has no intersection with $\mathcal{C}_s$ (data from $q_{\overline{\mathcal{C}}_t}$ does not belong to any class in $\mathcal{C}_s$), and thus the hypothesis $\mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}_t}} H(\hat{\mathbf{y}}) > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}}} H(\hat{\mathbf{y}})$ is reasonable.

With the above analysis, the sample-level transferability criterion for source data points and target data points can be respectively defined as Eq. (7) and Eq. (8):

$$w^s(\mathbf{x}) = \frac{H(\hat{\mathbf{y}})}{\log |\mathcal{C}_s|} - \hat{d}'(\mathbf{x}) \tag{7}$$

$$w^t(\mathbf{x}) = \hat{d}'(\mathbf{x}) - \frac{H(\hat{\mathbf{y}})}{\log |\mathcal{C}_s|} \tag{8}$$

Note that the entropy is normalized by its maximum value ($\log |\mathcal{C}_s|$) so that it is restricted into $[0, 1]$ and comparable to the domain similarity measure $\hat{d}'$. Also, the weights are normalized into interval $[0, 1]$ during training.

The proposed universal adaptation network (UAN) leverages the sample-level transferability criterion to disentangle source data in $\mathcal{C}$, $\overline{\mathcal{C}}_s$ and target data in $\mathcal{C}$, $\overline{\mathcal{C}}_t$. As such, the category gap is reduced. The domain gap is reduced as well by aligning features between domains in shared label set $\mathcal{C}$.

4. Experiments

To perform a thorough evaluation, we compare UAN with state of the art methods tailored to various domain adaptation settings under a variety of UDA settings on several datasets with different ξ, $|\mathcal{C}_s \cup \mathcal{C}_t|$, $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$. Then, we explore the performance with respect to the change of ξ, $|\mathcal{C}_s \cup \mathcal{C}_t|$, $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$. We further provide comprehensive analyses of the hyper-parameter sensitivity and the quality of sample-level transferability criterion about the proposed UAN model. Code and data will be available at `github.com/thuml`.

4.1. Experimental Setup

In this subsection, we describe the datasets, the evaluation protocols and the implementation details.

Table 1. Average class accuracy (%) of universal domain adaptation tasks on **Office-Home** ($\xi = 0.15$) dataset (ResNet)

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 [13]	59.37	76.58	87.48	69.86	71.11	81.66	73.72	56.30	86.07	78.68	59.22	78.59	73.22
DANN [6]	56.17	81.72	86.87	68.67	73.38	83.76	69.92	56.84	85.80	79.41	57.26	78.26	73.17
RTN [23]	50.46	77.80	86.90	65.12	73.40	85.07	67.86	45.23	85.50	79.20	55.55	78.79	70.91
IWAN [45]	52.55	81.40	86.51	70.58	70.99	85.29	74.88	57.33	85.07	77.48	59.65	78.91	73.39
PADA [45]	39.58	69.37	76.26	62.57	67.39	77.47	48.39	35.79	79.60	75.94	44.50	78.10	62.91
ATI [28]	52.90	80.37	85.91	71.08	72.41	84.39	74.28	57.84	85.61	76.06	60.17	78.42	73.29
OSBP [35]	47.75	60.90	76.78	59.23	61.58	74.33	61.67	44.50	79.31	70.59	54.95	75.18	63.90
UAN w/o d	61.60	81.86	87.67	74.52	73.59	84.88	73.65	57.37	86.61	81.58	62.15	79.14	75.39
UAN w/o y	56.63	77.51	87.61	71.96	69.08	83.18	71.40	56.10	84.24	79.27	60.59	78.35	72.91
UAN	**63.00**	**82.83**	**87.85**	**76.88**	**78.70**	**85.36**	**78.22**	**58.59**	**86.80**	**83.37**	**63.17**	**79.43**	**77.02**

Table 2. Average class accuracy (%) on **Office-31** ($\xi = 0.32$), **ImageNet-Caltech** ($\xi = 0.07$) and **VisDA2017** ($\xi = 0.50$) (ResNet)

Method	Office-31							ImageNet-Caltech		VisDA
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	I → C	C → I	
ResNet [13]	75.94	89.60	90.91	80.45	78.83	81.42	82.86	70.28	65.14	52.80
DANN [6]	80.65	80.94	88.07	82.67	74.82	83.54	81.78	71.37	66.54	52.94
RTN [23]	85.70	87.80	88.91	82.69	74.64	83.26	84.18	71.94	66.15	53.92
IWAN [45]	85.25	90.09	90.00	84.27	84.22	86.25	86.68	72.19	66.48	58.72
PADA [45]	85.37	79.26	90.91	81.68	55.32	82.61	79.19	65.47	58.73	44.98
ATI [28]	79.38	92.60	90.08	84.40	78.85	81.57	84.48	71.59	67.36	54.81
OSBP [35]	66.13	73.57	85.62	72.92	47.35	60.48	67.68	62.08	55.48	30.26
UAN	**85.62**	**94.77**	**97.99**	**86.50**	**85.45**	**85.12**	**89.24**	**75.28**	**70.17**	**60.83**

4.1.1 Datasets

Office-31 [33] is *de facto* for visual domain adaptation with 31 categories in 3 visually distinct domains (**A**, **D**, **W**). We use the 10 classes shared by **Office-31** and **Caltech-256** [7] as the common label set $\mathcal{C}$, then in alphabetical order, the next 10 classes are used as the $\overline{\mathcal{C}}_s$, and the reset 11 classes are used as the $\overline{\mathcal{C}}_t$. Here $\xi = 0.32$.

Office-Home [40] is a larger dataset with 65 object categories in 4 different domains: Artistic images (**Ar**), Clip-Art images (**Cl**), Product images (**Pr**) and Real-World images (**Rw**). In alphabet order, we use the first 10 classes as $\mathcal{C}$, the next 5 classes as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. Here $\xi = 0.15$.

VisDA2017 [29] dataset focuses on a special domain adaptation setting (simulation to real). The source domain consists of images generated by game engines and target domain consists of real-world images. There are 12 classes in this dataset. We use the first 6 classes as $\mathcal{C}$, the next 3 classes as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. Here $\xi = 0.50$.

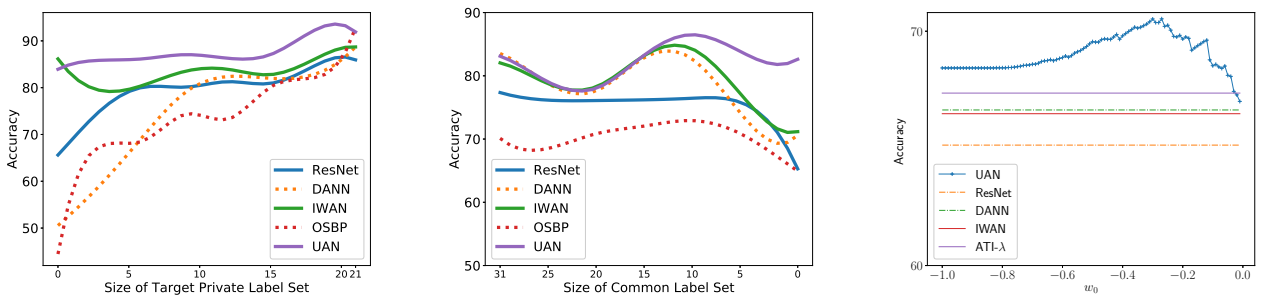
ImageNet-Caltech is built from **ImageNet-1K** [31] with 1000 classes and **Caltech-256** with 256 classes. As in previous works [2, 3], we used the 84 common classes shared by both domains as the common label set $\mathcal{C}$ and use their private classes as the private label set respectively. This dataset naturally falls into the universal domain adaptation paradigm. We form two universal domain adaptation tasks: **I** → **C** and **C** → **I**. Here $\xi = 0.07$.

These dataset settings are set up to both comply with the existing configurations [2, 3, 35, 28] and cover as many commonness levels ξ as possible, since brute-force evaluation of all combinations of ξ, $|\mathcal{C}_s \cup \mathcal{C}_t|$, $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$ is unacceptable.

4.1.2 Evaluation Details

Compared Methods. We compare the proposed UAN with (**1**) Convolutional Neural Network: **ResNet** [13], (**2**) close-set domain adaptation methods: Domain-Adversarial Neural Networks (**DANN**) [6], Residual Transfer Networks (**RTN**) [23], (**3**) partial domain adaptation methods: Importance Weighted Adversarial Nets (**IWAN**) [45], Partial Adversarial Domain Adaptation (**PADA**) [3], (**4**) open set domain adaptation methods: Assign-and-Transform-Iteratively (**ATI**) [28], Open Set Back-Propagation (**OSBP**) [35]. These methods are state of the art in their respective settings (ATI-λ is compared and λ is derived as described in [28]). It shall be valuable to study the performance of these methods in the practical UDA setting.

Evaluation Protocols. We adopt the evaluation protocol in Visual Domain Adaptation (VisDA2018) Open-Set Classification Challenge, where all the data in the target private label set is regarded as one unified "unknown" class and the average of per-class accuracy for all the $|\mathcal{C}| + 1$ classes is the final result. We extend existing methods by confidence thresholding. At the testing stage, if the prediction confi-



(a) Accuracy w.r.t. $|\overline{\mathcal{C}}_t|$ (b) Accuracy w.r.t. $|\mathcal{C}|$ (c) Sensitivity to w_0

Figure 3. (a) Accuracy w.r.t. $|\overline{\mathcal{C}}_t|$ in task **A** $\rightarrow$ **D**, $\xi = 0.32$. (b) Accuracy w.r.t. $|\mathcal{C}|$ in task **A** $\rightarrow$ **D**. (c) Performance w.r.t. threshold w_0.

dence is under the confidence threshold, the input image is classified as "unknown".

Implementation Details. Implementation is in PyTorch and ResNet-50 [13] is used as the backbone network. Models are fine-tuned from ResNet-50 pre-trained on ImageNet. We set temperature [14] as 10 when calculating $\hat{\mathbf{y}}$ in Eq. (7) because the prediction for source data is usually too certain and the entropy is low. When applied in Eq. (3), w^s, w^t are normalized in a mini-batch to be within interval $[0, 1]$.

4.2. Classification Results

The classification results are shown in Tables 1 and 2, respectively. UAN outperforms all the compared methods in terms of the average per-class accuracy. In particular, we have some key observations.

In the practical UDA setting, especially in the difficult Office-Home dataset, most existing methods perform similarly to or even worse than ResNet, indicating that existing methods are prone to negative transfer in UDA settings, meaning that they perform worse than a model only trained on source data without any adaptation. For example, Figure 4(a) shows the per-class accuracy gain compared to ResNet on task **Ar** $\rightarrow$ **Cl**. We can find that DANN, IWAN, and OSBP suffer from negative transfer in most classes and are only able to promote the adaptation for a few classes. Only UAN promotes positive transfer for all classes.

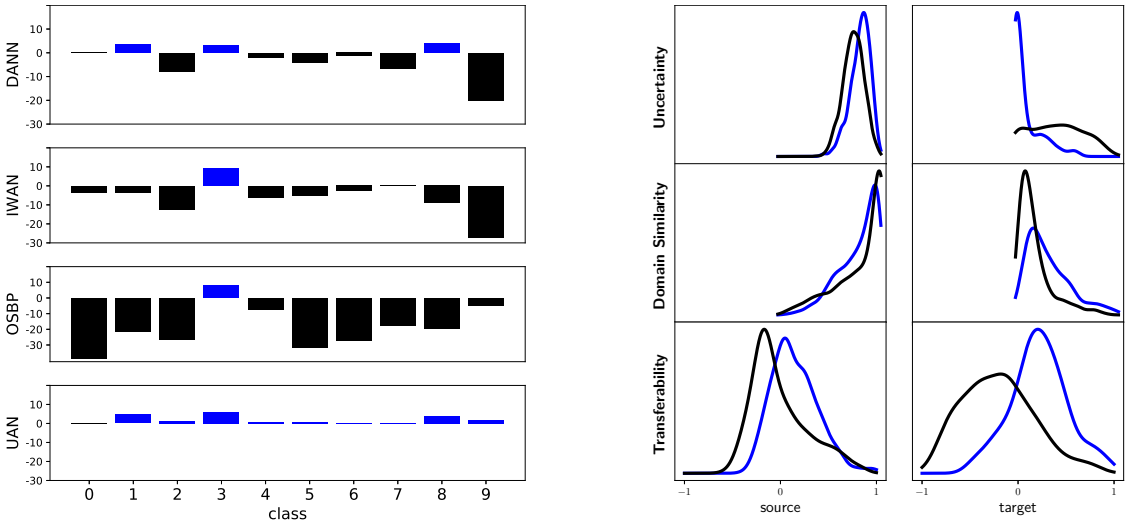
In these various settings, UAN outperforms all the mentioned methods. This is because UAN has a carefully designed sample-level transferability criterion. It filters out data coming from $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$ on feature alignment and provides a better criterion for "unknown" class detection than the existing confidence thresholding method.

Existing methods perform well when their assumptions hold but worse when violated. Take OSBP as an example, if manually removing source private classes (invalid operation since target labels are unknown), its accuracy is 89.1% on Office-31; however, if keeping source private classes (violating its assumption), its accuracy drops to 67.68%. As the assumptions of previous open set DA methods are violated in UDA, it is no wonder that their accuracies drop sharply.

4.3. Analysis on Different UDA Settings

Varying Size of $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$. With fixed $|\mathcal{C}_s \cup \mathcal{C}_t|$ and ξ, we explore the performance of methods mentioned above on universal domain adaptation with the various sizes of $\overline{\mathcal{C}}_t$ ($\overline{\mathcal{C}}_s$ also changes correspondingly) on task **A** $\rightarrow$ **D** in Office-31 dataset. As shown in Figure 3(a), UAN outperforms all the compared methods on most sizes of $\overline{\mathcal{C}}_t$. In particular, when $|\overline{\mathcal{C}}_t| = 0$, which is the partial domain adaptation setting with $\mathcal{C}_t \subset \mathcal{C}_s$, the performance of UAN is comparable to IWAN's performance. And when $|\overline{\mathcal{C}}_t| = 21$, which is the open set domain adaptation setting with $\mathcal{C}_s \subset \mathcal{C}_t$, the performance of UAN is comparable to OSBP's performance. IWAN and OSBP both take advantage of the prior knowledge about label sets and design modules to exploit the knowledge. However, UAN can still catch up with them in their expert settings, indicating UAN is effective and robust to diverse sizes of $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$. In the middle of 0 and 21, where $\mathcal{C}_s$ and $\mathcal{C}_t$ are partly shared, UAN outperforms other methods with large margin. UAN can produce impressive results without any prior knowledge about the target label set. The general trend in Figure 3(a) is that the performance goes higher when $|\overline{\mathcal{C}}_t|$ becomes larger. This is natural since larger $|\overline{\mathcal{C}}_t|$ means smaller $|\overline{\mathcal{C}}_s|$ and less distraction to the label classifier.

Varying Size of Common Label Set $\mathcal{C}$. We explore another dimension of universal domain adaptation by varying the size of $\mathcal{C}$. This is done in Office-31 dataset on task **A** $\rightarrow$ **D**. Here $|\mathcal{C}| + |\overline{\mathcal{C}}_t| + |\overline{\mathcal{C}}_s| = 31$. For simplicity, we let $|\overline{\mathcal{C}}_t| = |\overline{\mathcal{C}}_s| + 1$ and vary $|\mathcal{C}|$ from 0 to 31. Figure 3(b) shows the accuracy of these methods with different $|\mathcal{C}|$'s. When $|\mathcal{C}| = 0$, source domain and target domain have no overlap on label sets, i.e. $\mathcal{C}_t \cap \mathcal{C}_s = \emptyset$. We observe that UAN substantially outperforms all the compared methods with large margin, because they all assume that there is some common label set between source and target domains and cannot filter out target samples well when all the target samples are in the private label set $\overline{\mathcal{C}}_t$. When $|\mathcal{C}| = 31$, which is the closed set domain adaptation setting with $\mathcal{C}_s = \mathcal{C}_t$, we see that the performance of UAN is comparable with DANN's performance, indicating that the sample-level transferability criterion of



(a) Negative Transfer in UDA

(b) Hypotheses Quality (blue for *common* and black for *private*)

Figure 4. (a) The negative transfer influence in UDA (task **Ar** $\rightarrow$ **Cl**). (b) Justification of validity of hypotheses in Section 3.4.

UAN preserves useful samples and does not influence performance on the closed set domain adaptation setting. Note that when $|\mathcal{C}|$ keeps decreasing, the performance of DANN and IWAN drops rapidly and only UAN works stably.

4.4. Analysis of Universal Adaptation Network

Ablation Study. We go deeper into the efficacy of the proposed sample-level transferability criterion by performing an ablation study that evaluates variants of UAN. (1) **UAN w/o d** is the variant without integrating the domain similarity into the sample-level transferability criterion in Eq. (7) and Eq. (8); (2) **UAN w/o y** is the variant without integrating the uncertainty criterion into sample-level transferability criterion in Eq. (7) and Eq. (8). Results are shown in bottom rows of Table 1. UAN outperforms UAN w/o d and UAN w/o y, indicating both the domain similarity component and the uncertainty criterion component in the definition of $w^s(\mathbf{x})$, $w^t(\mathbf{x})$ are important and necessary. In addition, UAN w/o d performs better than UAN w/o y, meaning that integrating the uncertainty criterion into the sample-level transferability criterion is even more crucial.

Hypotheses Justification. To justify the validity of the hypotheses in Section 3.4, we plot in Figure 4(b) the estimated probability density function for different components of weights $w^s(\mathbf{x})$ in Eq. (7) and $w^t(\mathbf{x})$ in Eq. (8). Results show that all the hypotheses are successfully justified, explaining why UAN can perform well in various UDA settings. Another observation is that the uncertainty criterion and the domain similarity themselves can be used to distinguish all the examples from common label set and private label sets. By combining these two components we can obtain more distinguishable transferability criterion.

Threshold Sensitivity. We explore the sensitivity of UAN with respect to threshold w_0 in task **I** $\rightarrow$ **C**. As shown in Figure 3(c), though UAN's accuracies vary by about 2% w.r.t. w_0, it consistently outperforms the other methods by large margins in a wide range of w_0. Note that the baselines are fully tuned and their best accuracies are compared here.

5. Conclusion

In this paper, we introduce a novel Universal Domain Adaptation (UDA) setting, where no prior knowledge are required on the label set relationship between domains. We propose Universal Adaptation Network (UAN) with a well-designed sample-level transferability criterion to address UDA. A thorough evaluation shows that existing methods requiring prior knowledge on the relationship of label sets cannot work well in general UDA setting while the proposed UAN works stably and achieves state-of-the-art results.

In practice, if one wants to generalize a model to a new scenario, the proposed UAN can be a good candidate model. If UAN classifies most examples as "unknown", then domain adaptation in such a new scenario may well fail, and collecting labels will be indispensable. On the other hand, if UAN can generate labels for most examples, collecting labels for such a scenario are not necessary and domain adaptation will perform the work. That said, UAN can serve as a pilot study when we encounter a new domain adaptation scenario.

Acknowledgements

This work is supported by National Key R&D Program of China (No. 2017YFC1502003) and National Natural Science Foundation of China (61772299, 71690231, and 61672313).

References

[1] B. Bhushan Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, September 2018.

[2] Z. Cao, M. Long, J. Wang, and M. I. Jordan. Partial transfer learning with selective adversarial networks. In *CVPR*, June 2018.

[3] Z. Cao, L. Ma, M. Long, and J. Wang. Partial adversarial domain adaptation. In *ECCV*, pages 135–150, 2018.

[4] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *CVPR*, pages 7976–7985, 2018.

[5] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *TPAMI*, 34(3):465–479, 2012.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17:59:1–59:35, 2016.

[7] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[9] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2004.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330.

[11] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *ICCV*, volume 2, page 6, 2017.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network.

[15] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1994–2003, 2018.

[16] L. Hu, M. Kan, S. Shan, and X. Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *CVPR*, June 2018.

[17] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *ECCV*, September 2018.

[18] G. Kang, L. Zheng, Y. Yan, and Y. Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *ECCV*, September 2018.

[19] B. Konstantinos, S. Nathan, D. David, E. Dumitru, and K. Dilip. Unsupervised pixel–level domain adaptation with generative adversarial networks. In *CVPR*, pages 95–104, 2017.

[20] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *CVPR*, June 2018.

[21] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[22] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional domain adversarial network. In *NeurIPS*, 2018.

[23] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, pages 136–144, 2016.

[24] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *NeurIPS*, pages 165–177, 2017.

[25] F. Maria Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo. Autodial: Automatic domain alignment layers. In *ICCV*, Oct 2017.

[26] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *CVPR*, June 2018.

[27] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *TNNLS*, 22(2):199–210, 2011.

[28] P. Panareda Busto and J. Gall. Open set domain adaptation. In *ICCV*, Oct 2017.

[29] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, K. Saenko, X. Roynard, J.-E. Deschaud, F. Goulette, and T. L. Hayes. VisDA: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshops*, pages 2021–2026.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[32] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: Symmetric bi-directional adaptive gan. In *CVPR*, June 2018.

[33] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[34] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, June 2018.

[35] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. Open set domain adaptation by backpropagation. In *ECCV*, September 2018.

[36] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, June 2018.

[37] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

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

[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[40] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[41] R. Volpi, P. Morerio, S. Savarese, and V. Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, June 2018.

[42] X. Wang and J. Schneider. Flexible transfer learning under support and model shift. In *NeurIPS*, 2014.

[43] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5423–5432, 2018.

[44] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *ICLR*.

[45] J. Zhang, Z. Ding, W. Li, and P. Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, June 2018.

[46] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013.

[47] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, June 2018.

[48] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2242–2251, 2017.