https://doi.org/10.1007/s11263-021-01463-x

Deep CockTail Networks

A Universal Framework for Visual Multi-source Domain Adaptation

Ziliang Chen[1,2] · Pengxu Wei[1] · Jingyu Zhuang[1] · Guanbin Li[1] · Liang Lin[1]

Received: 30 January 2020 / Accepted: 23 March 2021 / Published online: 20 May 2021

Abstract
Transferable deep representations for visual domain adaptation (**DA**) provides a route to learn from labeled source images to recognize target images without the aid of target-domain supervision. Relevant researches increasingly arouse a great amount of interest due to its potential industrial prospect for non-laborious annotation and remarkable generalization. However, DA presumes source images are identically sampled from a single source while Multi-Source DA (**MSDA**) is ubiquitous in the real-world. In MSDA, the domain shifts exist not only between source and target domains but also among the sources; especially, the multi-source and target domains may disagree on their semantics (e.g., category shifts). This issue challenges the existing solutions for MSDAs. In this paper, we propose Deep CockTail Network (**DCTN**), a universal and flexibly-deployed framework to address the problems. DCTN uses a multi-way adversarial learning pipeline to minimize the domain discrepancy between the target and each of the multiple in order to learn domain-invariant features. The derived source-specific perplexity scores measure how similar each target feature appears as a feature from one of source domains. The multi-source category classifiers are integrated with the perplexity scores to categorize target images. We accordingly derive a theoretical analysis towards DCTN, including the interpretation why DCTN can be successful without precisely crafting the source-specific hyper-parameters, and target expected loss upper bounds in terms of domain and category shifts. In our experiments, DCTNs have been evaluated on four benchmarks, whose empirical studies involve vanilla and **three challenging category-shift transfer problems in MSDA**, i.e., source-shift, target-shift and source-target-shift scenarios. The results thoroughly reveal that DCTN significantly boosts classification accuracies in MSDA and performs extraordinarily to resist negative transfers across different MSDA scenarios.

Keywords Multi-source domain adaptation · Cross-domain visual recognition · Domain shift · Category shift · Open-set domain adaptation · Diverse transfer scenarios

1 Introduction

Considerable advances in deep representation learning have recently improved the state-of-the-art approaches on a huge variety of machine vision problems (Krizhevsky et al. 2012; Ren et al. 2015; Long et al. 2015; Liang et al. 2016; Xu et al. 2015; Johnson et al. 2017; Ho and Gopalan 2014; Kan et al. 2014; Zhang et al. 2019). These eyeball-catching prospects can be greatly attributed to the availability of large scale labeled datasets for supervised learning (Deng et al. 2009; Cordts et al. 2015). Nevertheless, these successes are challenged by domain shift problem (Pan and Yang 2010), since the traditional assumptions that their training dataset and test set follow the same distributions are often violated.

Communicated by Minsu Cho.

✉ Pengxu Wei
weipx3@mail.sysu.edu.cn

Ziliang Chen
c.ziliang@yahoo.com

Jingyu Zhuang
zhuangjy6@mail2.sysu.edu.cn

Guanbin Li
liguanbin@mail.sysu.edu.cn

Liang Lin
linliang@ieee.org

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

[2] Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA

This poses a major obstacle in adapting predictive models across domains and leads to a performance degradation on target domains (Gretton et al. 2009).

To mitigate the negative effects caused by domain shift, (unsupervised) Domain Adaptation (DA) (Pan and Yang 2010) arises to reduce the discrepancy between the source and target domain distributions, typically by exploring domain-invariant data structures or transferable representations, which endows the classifier with the consistent classification ability on source and target examples (Tzeng et al. 2015; Bousmalis et al. 2017; Gebru et al. 2017). Most existing DA approaches are preconditioned on a single source where labeled examples are identically drawn from an individual source underlying distribution. This setup is widely admitted in traditional DA researches, while merely reflects a tip of the iceberg of realistic transfer circumstances.

In a variety of real-world cases, we often witness data drawn from multiple source domains. For instance, for the sake of illness typicality, medical images are conventionally collected from hospitals all over the country in a long time. This application circumstances produce a large amount of datasets that should be treated as a set of multiple sources. Consequently, Multi-Source Domain Adaptation (MSDA) has increasingly grabbed considerable attention in many applications (Yang et al. 2007; Duan et al. 2012; Jhuo et al. 2013a), since reasonable approaches might achieve more transfer learning performance
gains.

Compared with the deep single-source DA with witnessed progresses, scarce researches have been committed to deep MSDA due to *complex domain shift conditions*. Especially, domain shift exists not only between a target and each source, but also across multiple source domains. MSDA presents in an extensive variety of scenarios arousing serious negative transfer influences (Pan and Yang 2010) due to the *category shifts* across domains. For instance, the categories distributed across multiple source domains may not guarantee their class consistencies (Fig. 1b). In this *source-category-shift MSDA* scenario, the shifts of multi-source domains and their categories should be taken into account. Another case is derived from the popular single-source open-set DA research (Busto and Gall 2017), where some part of categories in a target domain are not included in source domains. These “outlier” categories are traditionally unified as a negative class called “unknown” (Fig. 1c). In our paper, this MSDA-extended open-set setting is termed *target-category-shift MSDA* scenario. More generally, these two cases would simultaneously occur and leads to *source-target-category-shift* MSDA scenarios (Fig. 1d). All these category-shift cases deteriorate the domain-shift damages to most existing DA algorithms and are nontrivial to solve.

Attempting to overcome the domain-shift and category-shift challenges, in this paper, we present *Deep CockTail Network* (DCTN), a flexibly-deployed adversarial learning framework to address MSDA problems across diverse transfer scenarios. DCTN encapsulates category classifiers for multi-source domains respectively, then the *target category predictor* is formulated by integrating their category probabilistic predictions on a target example with their source-target perplexity scores. In particular, each of the perplexity scores represents the domain-feature similarity between the target and each source, thus, referring to the outcome each source-target domain discriminator produces (*source-target domain discriminators are deployed to separate features from the target and each source. Like other adversarial DA approaches, domain discriminators of DCTN facilitates to learn a domain-invariant feature extractor in a multi-source-domain condition*). The more similarity between a source and the target on their features, the more convincing this source-specific classifier predicts the target example. Hence each target feature would be fed into multi-source classifiers, whose predictions are reweighted by the source-specific perplexity scores to classify this target example. Analogous to make cocktail, it inspires our framework dubbed by Deep CockTail Network.

Theoretically, we compare DCTN with the methods based on *source distribution weighted combining rule* (Mancini et al. 2009), where the target distribution is supposed to be represented as the weighted linear combination of multiple source distributions. This old-school MSDA theory provides an adaptation upper bound of expected classification error on the target domain. However, the multi-source weight combination involves with crafting a set of source-specific hyper-parameters by experiences; thus, it could not keep abreast of the current advancing approaches. DCTN does not rely on this mixture assumption. In contrast, the learning algorithm of DCTN employs a multi-way adversarial scheme to adaptively decide the multi-source balancing rule according to their source-target perplexity scores. The DCTN target category prediction maintains an expected loss upper bound underlying a reasonable DA approximation presumption, yet free of specifying multi-source hyper-parameters. More importantly, it can be developed to suit the cases of source and target category shifts in MSDA.

Overall, our work mainly contributes in three aspects:

- We investigate four representative MSDA scenarios, i.e., *vanilla*, *source-category-shift*, *target-category-shift* and *source-target-category-shift*, and propose *Deep CockTail Network* (DCTN), to solve these challenging and complex MSDA problems by a universal framework.

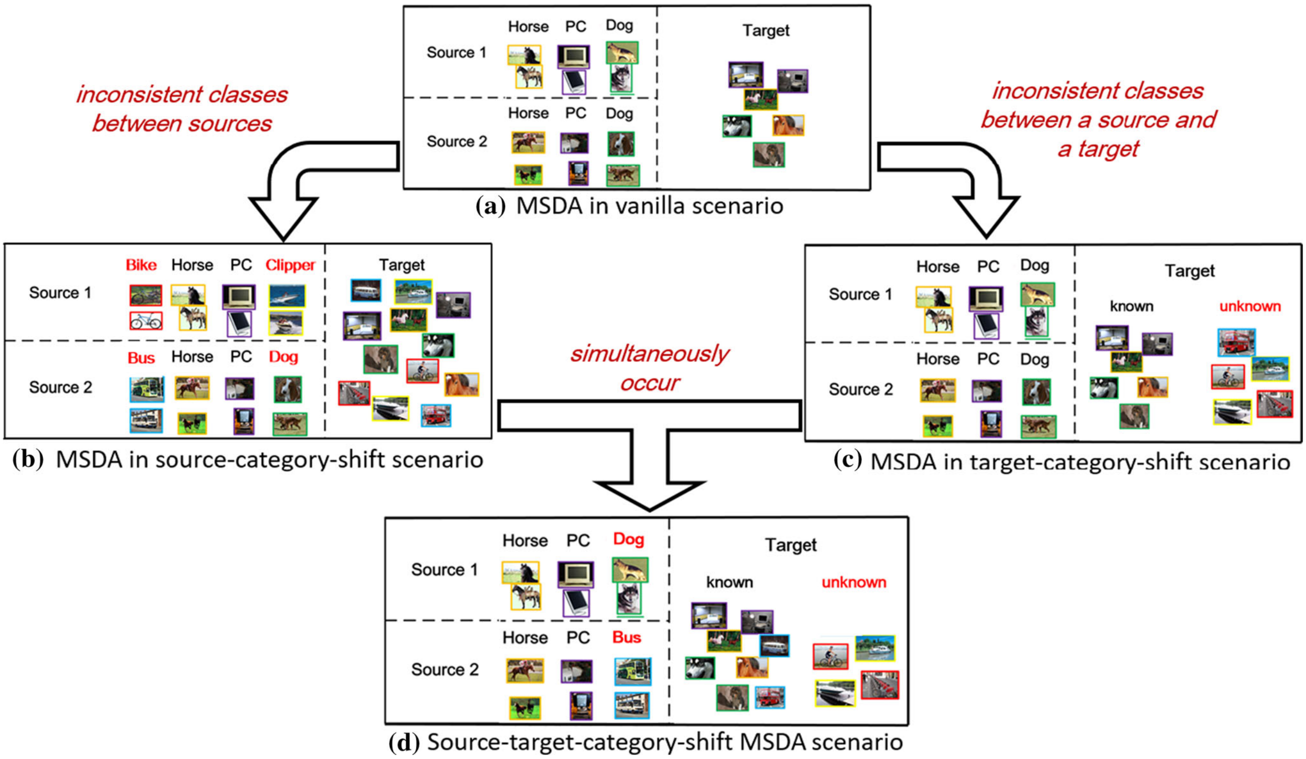
Source 1
Horse
PC
Dog
Source 2
Target
(a) MSDA in vanilla scenario
inconsistent classes between sources
inconsistent classes between a source and a target
Bike
Clipper
Bus
(b) MSDA in source-category-shift scenario
simultaneously occur
known
unknown
(c) MSDA in target-category-shift scenario
(d) Source-target-category-shift MSDA scenario

Fig. 1 A brief illustration of Multi-Source (unsupervised) Domain Adaptation (MSDA) scenarios and their hierarchical relation. **a** Vanilla MSDA scenarios consider multi-source and target examples that exactly share their categories. **b** Multi-source data are collected from source domains where domain shift and categorical misalignment co-exist between the source domains. **c** Multiple sources meet an open-set target domain (Saito et al. 2018) including some "unknown" categories non-existent in the sources. **d** Source and target category shifts (**b**, **c**) simultaneously occur in this scenario. Note that, **b** is derived from our original version (Xu et al. 2018); **c**, **d** are first taken into account in this paper. MSDA problems in these scenarios can be settled by our DCTN. (For simplicity, we only reveal the cases with two source domains. Best viewed in color.)

- Under two assumptions derived from our learning algorithm, we develop the bound of the target instance loss in DCTN. It explains why DCTN can success without relying on the source-distribution-weighted-combing rule. Based on this, we develop the upper bounds of target expected loss across all aforementioned MSDA scenarios.
- We conduct extensive experiments on four MSDA benchmarks including diverse source-to-target transfer cases in four different category shift scenarios. Our experimental results demonstrate the superiority and versatility of DCTN.

The remainder of this paper can be concluded as follows. Related works are described in Sect. 2. Details of problem setup on diverse MSDA problems are in Sect. 3 and our method is presented in Sect. 4. Experimental results are given in Sect. 5. We conclude this paper in Sect. 6.

2 Related Work

2.1 Domain Adaptation with a Single Source

Provided a source domain with ground truth and target domain without labels, unsupervised domain adaptation (Pan and Yang 2010; Gong et al. 2014; Shao et al. 2014; Xu et al. 2016) aims to learn a model well-performed on a target domain. Since the source and target belong to different distributions, the technical problem in UDA is how to mitigate the domain shift between them. Inspired by the two-sample test (Gretton et al. 2007), various statistical discrepancy measures can be directly applied to regulate the domain shift during optimization, e.g., shallow-model-based TCA (Pan et al. 2011), JDA (Baktashmotlagh et al. 2016), deep-model-based DAN (Long et al. 2015), CMD (Zellinger et al. 2017), WMMD (Yan et al. 2017), RTN (Long et al. 2016), STN (Yao et al. 2019), in which diverse statistical measures are used as the regularizer to learn domain-invariant features.

Adversarial learning behaves effectively to learn more transferable representations (Ganin et al. 2017; Tzeng et al. 2017). It determines a couple of networks and trains them in the opposite direction: a domain discriminator minimizes the classification error to distinguish samples from source and target, while domain mapping learns transferable representations by confusing the domain discriminator. These so-called adversarial DA algorithms are classed into three branches. The first alternatively trains discriminator and feature extractor so that the extractor is encouraged to directly confuse the source and target. Namely, the probabilistic discriminative decisions about learned transferable representations should be consistent with $[\frac{1}{2}, \frac{1}{2}]$, no matter which domain the examples come from. The second proposes a reversal gradient layer, which flips the gradient values after its back-propagated from the discriminator. The operation allows a joint learning of discriminator and feature extractor and is easy for implementation, which makes it very popular in the adversarial domain adaptation. Finally, GAN-style adversary (Goodfellow et al. 2014) also suits a domain adaptation setting (Tzeng et al. 2017), which mostly performs as an asymmetric transfer pipeline. Due to the flexibility of adversarial learning framework, recent researches about adversarial DA perform superiorly in visual recognition across domains (Long et al. 2016; Gebru et al. 2017) and tasks (Motiian et al. 2017) and transfer structure learning (Bousmalis et al. 2017; Hoffman et al. 2016).

Besides these mainstream branches of DA approaches, there are also other diverse methods to learn domain-invariant features: semi-supervised method (Saito et al. 2017), domain reconstruction (Ghifary et al. 2016), duality (Haeusser et al. 2017), alignments (Fernando et al. 2013; Zhang et al. 2017; Sun et al. 2016), manifold learning (Gong et al. 2012), tensor methods (Koniusz et al. 2017; Lu et al. 2017), feature norm adaptation (Xu et al. 2019), etc.

2.2 Multi-source Domain Adaptation

The UDA approaches mentioned above mainly consider target domain versus single source domain. If multiple sources are available, the domain shifts among sources should also be considered. A-SVM (Yang et al. 2007) leverages the ensemble of source-specific classifiers to fine-tune the target categorization model; The Domain Adaptation Machine (Duan et al. 2012) introduces domain-dependent regularizer term based on a smoothness assumption, and enforces target classifier to make a similar decision to the relevant source classifier. Domain reconstruction method (Jhuo et al. 2013a) enforces different source domains to have jointly low ranks, which forms a compact source set close to the target domain.

MSDA also develops with some theoretical supports (Mancini et al. 2009; Blitzer et al. 2008; Ben-David et al. 2010). Blitzer et al. (2008) firstly provide the learning bound for MSDA. Mancini et al. (2009) claims that an ideal target hypothesis can be represented by a distribution weighted combination of source hypotheses. This methodology is so-called *source distribution weighted combining rule*, closely meaning that if the relations between target and each source can be discovered, we are able to use multiple source-specific classifiers to obtain an ideal target class prediction.

Very recently, some approaches based on neural nets attempt to address the MSDA problem. Zhao et al. (2018) developed a new adversarial learning paradigm by iteratively constructing zero-sum games between the target and one of the source domains. Mancini et al. (2018) proposes a multi-DA normalization layer that aligns multi-source domains in the target. They indeed have facilitated the progresses in MSDA whereas presented several limitations. Inspired by Xu et al. (2018) and Peng et al. (2019) develops a discrepancy-based DA algorithm to reweight the importances of multiple domains. They have performed promising results in a vanilla MSDA problem but if category shifts simultaneously exist across domains, which commonly appears in practice, they become unavailable.

2.3 Category-Shift Problems in Domain Adaptation

Most existing DA literatures consider DA problems in a close-set DA setup, where source domain and target domain exactly share their categories. This transfer precondition simplifies the analysis of most DA algorithms, but is unable to handle the situation where source and target categories are different. Increasingly, a variety of researches in turn focus on addressing these more challenging problems. For examples, Kim et al. (2020) suggests a new DA paradigm to address potential data-leakage issues. Cao et al. (2018) and Cao et al. (2018) investigate the partial DA problem where target categories presents as a proper subset of the source categories. Distinctly, Saito et al. (2018) and Busto and Gall (2017) investigate the open-set DA problem where some of target categories are unknown in the source domain[1]. (You et al. 2019) investigates universal domain adaptation (UDA), i.e., the DA problem concluding the aforementioned two scenarios. These existing category-shift transfer scenarios, however, rely on a "single-source-domain" setting. In contrast, diverse category shifts usually appear in practical MSDA problems, which is the focus of this paper.

3 Overview of MSDA

In unsupervised domain adaptation, images from target domain lack of annotation, hampers a straightforward usage of supervised learning to acquire a classifier adaptive to target distribution. Source domain offers categories information via a circuitous route. Nevertheless, category-shift

[1] More precisely, Saito et al. (2018) and Busto and Gall (2017) consider two different open-set problems.

problem is aggravated in MSDA compared with single source domain adaptation. In this paper, we explore category-shift problem for MSDA and summarize four representative adaptation scenarios, i.e., *vanilla*, *source-category-shift*, *target-category-shift* and *source-target-category-shift* MSDAs. In the following section, we will elaborate these four adaptation scenarios in a principle way.

In the context of multi-source domain adaptation, source domain images $\{(\mathbf{X}_j, \mathbf{Y}_j)\}_{j=1}^{M}$ are drawn from M different source domains with underlying distributions $\{P_j(\boldsymbol{x}, \boldsymbol{y})\}_{j=1}^{M}$, respectively. $\mathbf{X}_j = \{\boldsymbol{x}_{ji}\}_{i=1}^{N_j}$ represents N_j images from source j in total and $\mathbf{Y}_j = \{\boldsymbol{y}_{ji}\}_{i=1}^{N_j}$ corresponds to their labels. Target domain images $\mathbf{X}^{(t)} = \{\boldsymbol{x}_i^{(t)}\}_{i=1}^{N^{(t)}}$ are drawn from underlying distribution $P_t(\boldsymbol{x}, \boldsymbol{y})$ without label observation $\mathbf{Y}^{(t)}$. For MSDA, images from source and target domains are utilized for training; test images $(\mathbf{X}_{test}, \mathbf{Y}_{test})$ are only drawn from the target to evaluate the classifier adaptation performance.

3.1 Vanilla MSDA

$\mathscr{C}_j$ denotes a category set of labels in $\mathbf{Y}_j$ for source domain j, $\mathscr{C}^{(s)}$ denotes the category set of the source domain, and $\mathscr{C}^{(t)}$ is the unobserved category set of our target domain. In vanilla MSDA scenario, the category sets of multiple sources ($\{\mathscr{C}_k\}_{k=1}^{M}$) and the target ($\mathscr{C}^{(t)}$) are consistent, namely, $\mathscr{C}^{(s)} = \mathscr{C}_k$ ($\forall k \in [M] = \{1, 2, \ldots, M\}$) and $\mathscr{C}^{(t)}$ holds $\mathscr{C}^{(s)} = \mathscr{C}_k = \mathscr{C}^{(t)}$. This scenario presumes M source and target domains customized by consistent category semantics.

3.2 Category-Shift Scenarios in MSDA

In a vanilla scenario, images darwn from different domains share the same category set. Distinguished from this, category-shift MSDA scenario advocates that the category sets from different domains maybe also different. To this end, $\mathscr{C}^{(s)} = \mathscr{C}_k = \mathscr{C}^{(t)}$ ($\forall\ k \in [M]$) is generalized to suit different scenarios.

Specifically, in the *source-category-shift* scenario, $\mathscr{C}^{(s)} = \mathscr{C}_k = \mathscr{C}^{(t)}$ ($\forall\ k \in [M]$) turns to $\mathscr{C}_j \neq \mathscr{C}_k$ and $\mathscr{C}^{(t)} = \bigcup_{j=1}^{M} \mathscr{C}_j$. $\mathscr{C}_j \cap \mathscr{C}_k$ indicates the public classes between sources j and k. $\mathscr{C}_j \cap \mathscr{C}_k \neq \mathscr{C}_j \cup \mathscr{C}_k$ refers to the *source category shift*. In the *target-category-shift* scenarios, $\mathscr{C}^{(s)} = \bigcup_{j=1}^{M} \mathscr{C}_j = \mathscr{C}^{(t)}$ becomes $\mathscr{C}^{(s)} = \bigcup_{j=1}^{M} \mathscr{C}_j \neq \mathscr{C}^{(t)}$. Resumbling the spirit of the recent open-set single-source DA researches (Busto and Gall 2017; Saito et al. 2018), we consider a target-category-shift MSDA scenarios holding $\mathscr{C}^{(s)} = \bigcup_{j=1}^{M} \mathscr{C}_j \subset \mathscr{C}^{(t)}$. The categories in $\mathscr{C}^{(t)} / \bigcup_{j=1}^{M} \mathscr{C}_j$ are conventionally unified and treated as an "unknown" category c_u. The MSDA is supposed to preclude the target examples belonging to $c_u = \mathscr{C}^{(t)} / \bigcup_{j=1}^{M} \mathscr{C}_j$ and correctly categorize the rest into $\{\mathscr{C}_j\}_{j=1}^{M}$.

Finally, in *source-target-category-shift* scenario, the aforementioned category shifts simultaneously occur. This encourages us to address the both challenges by a unified framework.

4 Deep CockTail Networks (DCTNs)

Irrespective of either vanilla or the other multi-source transfer scenarios, MSDAs remain challenging to tackle and moreover, few researches are investigated under deep DA background. In this section, we introduce *Deep CockTail Network* (DCTN), an adversarial domain adaptation framework specified for MSDA. The framework is tailor-designed to address a vanilla MSDA problem yet ought to be noted that, DCTN could also be flexibly deployed to adapt the target domains in the source-category-shift, target-category-shift and source-target-shift-open-set scenarios by a mildly reconfiguring the learning pipeline.

In Sect. 4.1, we elaborate the basic pipeline of DCTN and the principle how DCTN predicts target data categories in diverse scenarios. In Sect. 4.2, we present the alternating learning algorithm of DCTN. In Sect. 4.3, we unveil the theoretical insight behind DCTN.

4.1 Framework

DCTN consists of four components: three subnets, i.e., *feature extractor*, *(multi-source) domain discriminator*, *(multi-source) category classifier*, and an unlearnable *cocktail target category predictor* to classify target examples (Fig. 2).

Feature extractor F incorporates a deep convolution neural network as the backbone, and maps images from M sources and the target into a common feature space. Apart from weight-shared architecture for all the domains, the joint adversarial learning with subsequent domain discriminators contributes F to learning both target-source-specific relations and domain-invariant features.

(Multi-source) domain discriminator D is built upon M source-specific discriminators $\{D_j\}_{j=1}^{M}$ for adversary. Given an image $\boldsymbol{x}$, the domain discriminator D receives the feature $F(\boldsymbol{x})$, and then each source-specific discriminator D_j ($\forall j \in [M]$) classifies respectively whether x originates from the source j or the target. The data flow coming from each source does not trigger those discriminators belonging to other source domains, while for the data flow from each target instance $\boldsymbol{x}^{(t)}$, the domain discriminator D yields M source-specific discriminative outcomes $\{D_j(F(\boldsymbol{x}^{(t)}))\}_{j=1}^{M}$ between the target and the M sources, respectively. They are leveraged to update the discriminator D, and provide the target-source

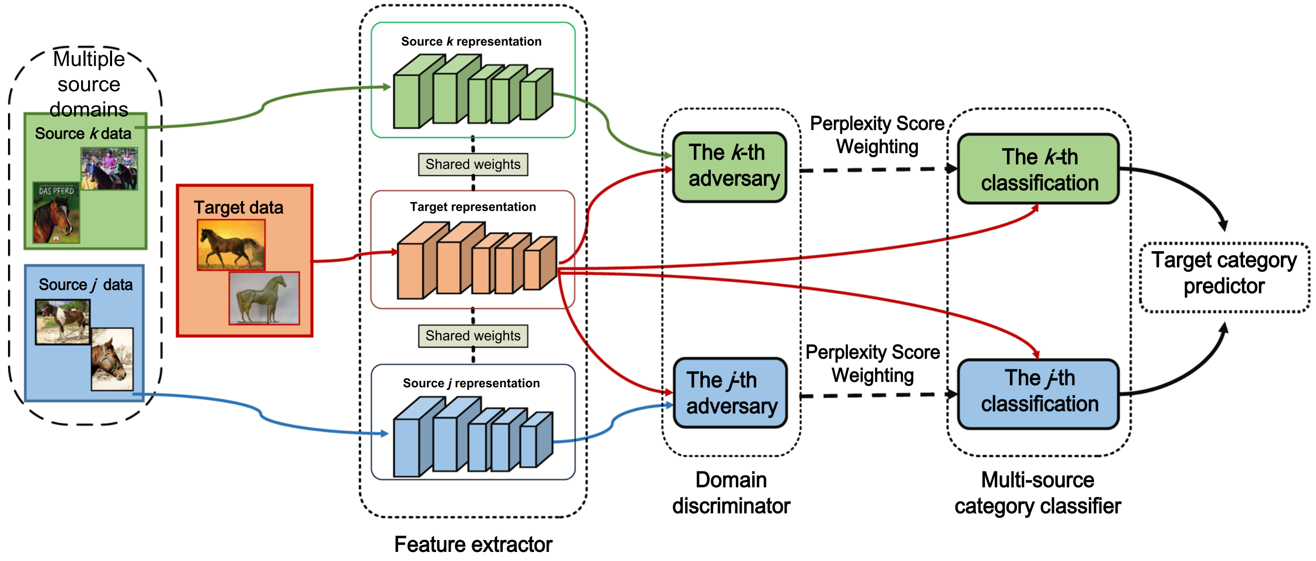

Fig. 2 The overview of Deep CockTail Network (DCTN). Our framework receives multi-source instances with annotated ground truth and adapts to classify the target samples. We confine the problem with only source j and k for simplicity. (**i**) The feature extractor maps target, source j and k into a common feature space. (**ii**) The category classifier receives target feature and produces the jth and kth classifications based upon the categories in source j and k respectively. (**iii**) The domain discriminator receives features from source j, k and target, then offers the kth advesary between target and source k, as well as the jth advesary between target and source j. The jth and kth advesary provide the source-j,k perplexity scores to reweight the jth and kth classifications correspondingly. (**iv**) The cocktail target category predictor integrates all reweighted classification results, so as to predict the target examples' categories across diverse category-shift scenarios. Since the samples from multi-source domains merely produce the training losses, their flows (outputs) are omitted to mainly illustrate the whole process of identifying target samples for simplicity. (Best viewed in color.)

perplexity scores $\{s(\boldsymbol{x}^{(t)}; F, D_j)\}_{j=1}^{M}$ defined as

$$s(\boldsymbol{x}^{(t)}; F, D_j) = -\log(1 - D_j(F(\boldsymbol{x}^{(t)}))). \tag{1}$$

$s(\boldsymbol{x}^{(t)}; F, D_j)$ implies how similar $\boldsymbol{x}^{(t)}$ is as a sample drawn from source j.

(Multi-source) category classifier C is a multi-output net composed of M source-specific category classifiers $\{C_j\}_{j=1}^{M}$. Each classifier is a softmax classifier configured by the category set that corresponds to its source. The category classifier takes an image mapping from feature extractor as input. For each image from source j, only the gradient derived from C_j is activated for the parameter updating. For a target image $\boldsymbol{x}^{(t)}$ instead, all source-specific classifiers provide M categorization results $\{C_j(F(\boldsymbol{x}^{(t)}))\}_{j=1}^{M}$, contributing to the parameters updating of C.

Cocktail target category predictor is the key component to categorize target examples. Specifically, given a target sample $\boldsymbol{x}^{(t)}$, our cocktail target category predictor takes its source perplexity scores $\{s(\boldsymbol{x}^{(t)}; F, D_j)\}_{j=1}^{M}$ to re-weight $\{C_j(F(\boldsymbol{x}^{(t)}))\}_{j=1}^{M}$, and then integrates them for classification. Here we specify the classification principle in four MSDA scenarios:

1. **In a vanilla MSDA scenario**, the category sets of a target domain and M source domains are consistent. Hence target category predictor is formulated by re-weighting source-specific classifier prediction with target-source perplexity scores:

$$C_t(\boldsymbol{x}^{(t)}) \;:=\; \sum_{j=1}^{M} \frac{s(\boldsymbol{x}^{(t)}; F, D_j)}{\sum_{k=1}^{M} s(\boldsymbol{x}^{(t)}; F, D_k)}\, C_j\big(F(\boldsymbol{x}^{(t)})\big), \tag{2}$$

where $C_t(\boldsymbol{x}^{(t)})$ denotes category probability forecast by the target predictor, and each entry of $C_t(\boldsymbol{x}^{(t)})$ denotes the integrated probability of $\boldsymbol{x}^{(t)}$ belonging to a specific category.

2. **In category-shift MSDA scenarios**, categories across M source and a target domains are not always shared. Therefore, we modify Eq. 2 to suit all cases. Specifically, each source classifier is obligated to classify those categories in its corresponding source. DCTN is expected to identify "unknown" c_u excluded by categories of M-source domains. To this, we activate all the source classifiers to identify the target examples in c_u. So Eq. 2 turns to

$$C_t(c|\boldsymbol{x}^{(t)}) \;:=\; \sum_{\mathscr{C}_j^{(t)}} \frac{s(\boldsymbol{x}^{(t)}; F, D_j)}{\sum_{\mathscr{C}_k^{(t)}} s(\boldsymbol{x}^{(t)}; F, D_k)} C_j\big(c|F(\boldsymbol{x}^{(t)})\big), \tag{3}$$

where $C_j\big(c|F(\boldsymbol{x}^{(t)})\big)$ represents the softmax prediction of the j^{th} source classifier $C_j\big(F(\boldsymbol{x}^{(t)})\big)$, i.e., the probability that $\boldsymbol{x}^{(t)}$ belongs to c.

$\mathscr{C}_j^{(t)}$ derived from $\mathscr{C}^{(t)}$ in Eq. 2 further includes the categories that M sources do not contain, e.g., $\mathscr{C}_j^{(t)} = \mathscr{C}_j \cup \{c_u\}$. So $C_j\big(F(\boldsymbol{x}^{(t)})\big)$ turns into a $(|\mathscr{C}_j| + 1)$-slot softmax category predictor, in order to synchronically recognize the categories in $\mathscr{C}_j$ and the "unknown" c_u. Note that only the sources including c would join the perplexity re-weighting to classify c.

4.2 Learning

DCTN follows an alternative adaptation pipeline given a pre-trained feature extractor and category classifier. At the very beginning of learning, we adopt source images to train the feature extractor F and the category classifier C. Those networks and the cocktail target classifier then predict categories for all target images[2] and annotate those with high confidences. Thus, we obtain the pre-trained feature extractor and category classifiers via fine-tuning with labeled multi-source images and pseudo-labeled target images. With pre-training, DCTN employs a multi-way adversary scheme to learn a mapping shared by all domains; then the feature extractor and the category classifiers are jointly trained with multi-source labeled and target pseudo-labeled images. These two stages repeat until the maximal epoch is reached.

4.2.1 Multi-way Adversarial Adaptation

Multi-way adversarial adaptation in DCTN is proposed to obtain domain-invariant features. It is formulated as follows:

$$\min_F \max_D V(F, D; C) = \mathscr{L}_{adv}(F, D) + \mathscr{L}_{cls}(F, C), \tag{4}$$

where the first term denotes our multi-way adversarial loss and the second term indicates cross-entropy losses for source-specific classification; C are frozen to offer stable values in the gradients. This multi-way adversarial loss are defined as

$$\begin{aligned}\mathscr{L}_{adv}(F, D) = \frac{1}{M}\sum_{j}^{M}\Big[&\mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_j}[\log D_j(F(\boldsymbol{x}))] \\ &+ \mathbb{E}_{\boldsymbol{x}^{(t)}\sim\mathbf{X}_t}[\log(1 - D_j(F(\boldsymbol{x}^{(t)})))]\Big].\end{aligned} \tag{5}$$

The optimization based on Eq. 5 is solely used to train D. Since the feature extractor F learns the mapping with respect to the multiple source domains and the target domain, the domain distributions simultaneously changes in adversary, which results in an oscillation that spoils our feature extractor. Regarding such concern, when source and target feature mappings share their architectures, the domain confusion can be introduced to substitute the adversarial objective (Tzeng et al. 2017), which performs stably to learn F. Inspired by this, we have obtained the multi-domain confusion loss:

$$\begin{aligned}\mathscr{L}_{adv}(F, D) = \frac{1}{M}\sum_{j}^{M}\Big[&\mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_j}\mathscr{L}_{cf}(\boldsymbol{x}; F, D_j) \\ &+ \mathbb{E}_{\boldsymbol{x}^{(t)}\sim\mathbf{X}_t}\mathscr{L}_{cf}(\boldsymbol{x}^{(t)}; F, D_j)\Big],\end{aligned} \tag{6}$$

where

$$\begin{aligned}\mathscr{L}_{cf}(\boldsymbol{x}; F, D_j) = &\frac{1}{2}\log D_j(F(\boldsymbol{x})) \\ &+ \frac{1}{2}\log(1 - D_j(F(\boldsymbol{x}))).\end{aligned} \tag{7}$$

Thus, DCTN updates its feature extractor F by optimizing the objective Eq. 4 w.r.t. Eq. 6.

Online hard source-domain batch mining When sampling mini-batches, the multi-way adversarial adaptation stochastically receives m examples from M sources respectively to update the feature extractor F in each iteration. However, the images drawn from different source domains would be not always helpful for boosting the adaptation, and as the model training proceeds, redundant source images would turn to draw back the previous adaptation performance. Thus we design a simple yet effective hard domain batch mining technique to improve the training efficiency. Specifically, in each iteration, DCTN randomly draws m target examples $\{\boldsymbol{x}_i^{(t)}\}_{i=1}^m$ and m source examples for each source, i.e., $\{\{\boldsymbol{x}_{1,i}\}_{i=1}^m, \cdots, \{\boldsymbol{x}_{M,i}\}_{i=1}^m\}$. So there are totally $m(M + 1)$ images for training DCTN per iteration. We keep the discriminator training as described in Eq. 4. As for feature extractor training, we independently consider the adversary between the target and each source. Given this, $\sum_i^m\{-\log D_j(F(\boldsymbol{x}_{j,i})) - \log[1 - D_j(F(\boldsymbol{x}_i^{(t)}))]\}$ can be viewed as the "difficulty" degree to distinguish $\boldsymbol{x}_i^{(t)}$ from m images of the source j. Therefore, if F performs worst to transform target features to confuse the source j^*, it leads to $j^* = \arg\max_{j\in[M]} \sum_i^m\{-\log D_j(F(\boldsymbol{x}_{ji})) - \log[1 - D_j(F(\boldsymbol{x}_i^{(t)}))]\}$. Based upon the domain confusion loss, we use the source j^* and the target examples in the mini-batch to train the feature extractor F. This technique is concluded by Algorithm 1.

4.2.2 Target Discriminative Adaptation

Resembling the spirit of existing work about adversarial DAs, the multi-way adversarial adaptation process does not con-

[2] Since the domain discriminator hasn't been trained, we take the uniform distribution simplex weight as the perplexity scores.

Algorithm 1 Mini-batch Learning via online hard source-domain batch mining

```
Input: Mini-batch $\{\boldsymbol{x}_i^{(t)}, \{\boldsymbol{x}_{ji}, \boldsymbol{y}_{ji}\}_{j=1}^{M}\}_{i=1}^{m}$ sampled from $\mathbf{X}_t$ and
$\{(\mathbf{X}_j, \mathbf{Y}_j)\}_{j=1}^{M}$ respectively; feature extractor $F$; domain discriminator $D$; category classifier $C$.
Output: Updated $F'$.
Determine source $j^* \in [M]$, where
    $j^* = \arg\max_{j} \sum_i^m -\log D_j(F(\boldsymbol{x}_{ji})) - \log[1 - D_j(F(\boldsymbol{x}_i^{(t)}))]$;
Compute $\mathscr{L}_{adv}^{j^*} = \sum_i^m \mathscr{L}_{cf}(\boldsymbol{x}_{j^*,i}; F, D_{j^*}) + \mathscr{L}_{cf}(\boldsymbol{x}_i^{(t)}; F, D_{j^*})$;
Replace $\mathscr{L}_{adv}$ in Eq. 4 with $\mathscr{L}_{adv}^{j^*}$ and update $F$ by Eq. 4.
return $F' = F$.
```

Algorithm 2 Learning algorithm for DCTN

```
Input: $N$ source labeled datasets $\{\mathbf{X}_j, \mathbf{Y}_j\}_{j=1}^{M}$; target unlabeled dataset $\mathbf{X}_t$; initiated feature extractor $F$; multi-source category classifier $C$ and domain discriminator $D$; confidence threshold $\gamma$; entropy threshold $\zeta$; maximal adversarial iteration $\beta$.
Output: well-trained feature extractor $F^*$, domain discriminator $D^*$ and multi-source category classifier $C^*$.
Pre-train C and F
while not converged do
    Multi-way Adversarial Adaptation:
    for 1:$\beta$ do
        Sample mini-batch from $\{\mathbf{X}_j\}_{j=1}^{M}$ and $\mathbf{X}_t$;
        Update $D$ by Eq. 4;
        Update $F$ by Algorithm 1;
    end for
    Target Discriminative Adaptation:
    Samples $\mathbf{X}_t^{(p)} \subset \mathbf{X}_t$ with pseudo labels $\mathbf{Y}_t^{(p)}$;
    Update $F$ and $C$ by Eq. 9.
end while
return $F^* = F$; $C^* = C/C^R$; $D^* = D$.
```

sider the category variation during learning. No matter which MSDA scenario is considered, DCTN undergoes the identical adversarial process. Though it is able to produce domain-invariant features, it does not insure their abilities to classify a target domain. Ben-David et al. (2010) demonstrate that, to accommodate a source classifier in the target, DA algorithms requires the category classifier working well on different domains. But in case of a variety of MSDA scenarios, their classifiers should account for the categorical mis-alignment across M sources and the target, to prevent the damage caused by the non-consistent category sets across M sources and the unobserved categories ("unknown") in the target domain.

To achieve a universal target category predictor, we incorporate target examples to learn classifiable features with source data via discriminatively fine-tuning $\{C_j\}_{j=1}^{M}$. We develop a switchable strategy to select and annotate target samples. To this, the feature extractor F and multi-source classifiers $\{C_j\}_{j=1}^{M}$ are trained with multi-source labeled samples and these pseudo-labeled target examples. In particular, we use the target category predictor obtained in the previous iteration to annotate each target sample. Afterwards, the strategy selects suitable target examples $\mathbf{X}_t^{(p)}$ and annotate them with pseudo labels $\mathbf{Y}_t^{(p)}$.

Specifically, DCTN incorporates two criteria to identify target samples with high confidences and low uncertainties, respectively. First, for each target sample, DCTN considers the category with the highest prediction probability according to Eq. 3. If the probability is larger than a threshold $\gamma > 0$, this target sample would be selected as a high-confidence candidate. Second, DCTN further takes the classfication entropy of Eq. 3 to identify the candidates with low uncertainties. In vanilla and source-category-shift scenarios, only the target samples with high confidences and low uncertainties would get pseudo labels and join the fine-tuning. In the scenarios with target category shifts, DCTN additionally incorporate the target samples with low uncertainties and categorize them into the unknown class c_u. The pseudo-labeling strategy is summarized as follows

$$\boldsymbol{y}^{(t)} = \begin{cases} \mathbf{1}_{c=\underset{\mathscr{C}_s \cup \{c_u\}}{\arg\max}\{C_t(c|\boldsymbol{x}^{(t)})\}} & \begin{array}{c}\mathrm{Ent}(C_t(\boldsymbol{x}^{(t)})) < \zeta \quad and \\ C_t(c|\boldsymbol{x}^{(t)}) > \gamma\end{array} \\ \mathbf{1}_{c_u} & \mathrm{Ent}(C_t(\boldsymbol{x}^{(t)})) \geq \zeta, \end{cases} \tag{8}$$

where $\mathbf{1}_c$ denotes the one-hot representation of the label w.r.t. the category c. $\mathrm{Ent}(C_t(\boldsymbol{x}^{(t)}))$ denotes the target category prediction entropy of $\boldsymbol{x}^{(t)}$. If $\boldsymbol{x}^{(t)}$ does not satisfy Eq. 8, it would be ignored in the discriminative adaptation phase. The discriminative adaptation objective is defined as

$$\begin{aligned} \min_{F,\, C} \mathscr{L}_{cls}(F, C) = & \sum_{j}^{M} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim(\mathbf{X}_j,\mathbf{Y}_j)}\Big[\mathscr{L}(C_j(F(\boldsymbol{x})), \boldsymbol{y})\Big] \\ & + \mathbb{E}_{(\boldsymbol{x}^{(t)},\hat{\boldsymbol{y}}^{(t)})\sim(\mathbf{X}_t^{(p)},\mathbf{Y}_t^{(p)})} \\ & \left[\sum_{j=1}^{M} \mathscr{L}(C_j(F(\boldsymbol{x}^{(t)})), \hat{\boldsymbol{y}}^{(t)})\right], \end{aligned} \tag{9}$$

where $\mathscr{L}$ denotes the cross-entropy loss between predictions and (pseudo) labels; $(\mathbf{X}_t^{(p)}, \mathbf{Y}_t^{(p)})$ represent the selected target data and their pseudo labels, which are leveraged to update F and $\{C_j\}_{j=1}^{M}$.

The hyper-parameters γ and ζ are very important in our annotation strategy. For γ, DCTN incorporates the threshold close to 1 in order to ensure the high prediction probability to the pseudo label of each selected target example during training. While it does not suit ζ. Concretely, after the first iteration in the alternative learning, some target images would be selected as belonging to the unknown classes, which further join to fine-tune the feature extractor, enabling each source-classifier to identify the unknown target classes. As the unknown classes can be gradually identified by our source-classifiers, their entropy value would become lower than those in the initial stage. A fixed threshold is not able to detect this change. To this, if DCTN uses a low entropy threshold ζ, the scheme will mistreat more known-class images as unknown classes; while if DCTN uses a high entropy threshold, the number of selected unknown-class target images would progressively decrease, even leading to no images selected into the unknown class for discriminative adaptation in the later stages. The both cases would harms the final performance of DCTNs. To overcome this problem, DCTN tend to choose a top x%. It implies that ζ is a virtual threshold and no matter how the training progresses, the uncertain target samples will be selected in a certain number. The detailed setup of γ and ζ is found in the "Appendix".

4.3 Theoretical Analysis

In this section, we dive deeper into DCTN from a theoretical perspective.

We first provide some notation and a brief introductin of distribution weight combining rule, which our method is inspired from. However, we find it needs to craft some source-specific hyper-parameters and is insuitable when neural networks are basic models. Therefore, we develop a new methodology connected with DCTN. By choosing more appropriate assumptions, it derives the learning adaptation upperbounds with regards to different MSDA problems.

4.3.1 Reviewing Distribution Weighted Combining Rule

Let $\{\mathscr{P}_j\}_{j=1}^M$ and $\mathscr{P}_t$ denote source and target distributions[3], respectively. Given an instance $\boldsymbol{x}$, $\{\mathscr{P}_j(F(\boldsymbol{x}))\}_{j=1}^M$ and $\mathscr{P}_t(F(\boldsymbol{x}))$ denote the probabilities of $\boldsymbol{x}$ drawn from $\{\mathscr{P}_j\}_{j=1}^M$ and $\mathscr{P}_t$, respectively. Following *source distribution weighted combining rule* (Mancini et al. 2009), the target distribution denotes a mixture of multi-source distributions with the coefficients by normalized source distributions weighted by an implicit simplex $\triangle = \{\lambda : \lambda_j \geq 0, \sum_{j=1}^M \lambda_j = 1\}_{j=1}^M$, namely, $\mathscr{P}_t(F(\boldsymbol{x})) = \sum_{c\in\mathscr{C}_k}^M \lambda_k \mathscr{P}_k(F(\boldsymbol{x}))$. For simplicity, we consider the vanilla MSDA case so that $\mathscr{C}^{(t)} = \mathscr{C}_k$, $\forall k \in [M]$. Under the assumption in Mancini et al. (2009), an ideal target classifier $C_t(c|\boldsymbol{x}^{(t)})$ is derived by integrating source classifiers $\{C_j(c|F(\boldsymbol{x}^{(t)}))\}_{j=1}^M$:

$$C_t(c|\boldsymbol{x}^{(t)}) = \sum_{j=1}^{M} \frac{\lambda_j \mathscr{P}_j(F(\boldsymbol{x}^{(t)}))}{\mathscr{P}_t(F(\boldsymbol{x}^{(t)}))} C_j(c|F(\boldsymbol{x}^{(t)})). \tag{10}$$

Therefore we frame it into DA theory to further give its interpretation. In more specific, $\mathscr{X}$ represents the input (feature) space; $f : \mathscr{X} \rightarrow \mathbb{R}$ denotes the target function to learn (refer to the labels); $h : \mathscr{X} \rightarrow \mathbb{R}$ denotes the hypotheses in $\mathscr{H}$ with respect to a specific underlying distribution (correspond to the classifier); $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ denote a classification loss function. The MSDA learning objective function is formulated as

$$\min_{h\in\mathscr{H}} \mathscr{L}(\mathscr{P}, h, f) = \mathbb{E}_{x\sim\mathscr{P}}[L(h(x), f(x))], \tag{11}$$

where according to the definition of $\{\mathscr{P}_j\}_{j=1}^M$ and $\mathscr{P}_t$, x denote the feature of $\boldsymbol{x}$ that $x \sim \mathscr{P}(F(\boldsymbol{x}))$. We would like to write it as $x \sim \mathscr{P}(x)$ in our analysis. Suppose M source hypotheses $\{h_1, \cdots, h_M\}$ correspond to $\{\mathscr{P}_1, \cdots, \mathscr{P}_M\}$ and thus, for all $j \in [M]$, $\mathscr{L}(\mathscr{P}, h, f) \leq \epsilon$, $(\epsilon > 0)$, source distribution weight combining rule holds an upper bound of target expected loss:

Proposition 1 (Mancini et al. 2009) *Given a target distribution $\mathscr{P}_t$ as a mixture P_λ of multiple source distributions $\{\mathscr{P}_j\}_{j=1}^M$ w.r.t.,λ, the expected loss of its mixture hypothesis h_λ is at most ϵ w.r.t. any target function f, i.e., $\mathscr{L}(P_\lambda, h_\lambda, f) \leq \epsilon$, where*

$$\mathscr{P}_t(x) = P_\lambda(x) = \sum_{j=1}^{M} \lambda_j \mathscr{P}_j(x), \ \forall\lambda \in \triangle, \tag{12}$$

and

$$h_\lambda(x) = \sum_{j=1}^{M} \frac{\lambda_j \mathscr{P}_j(x)}{\mathscr{P}_t(x)} h_j(x). \tag{13}$$

The mixture hypothesis h_λ corresponds to $C_t(c|\boldsymbol{x}^{(t)})$ in Eq. 10. The theorem demonstrates that, if we are able to find optimal hyper-parameters $\lambda \in \triangle$, the target distribution can be represented as a mixture of multiple source distributions in Eq. 12, and the target classifier is certified to keep an upper bound of target distribution.

Equation 10 becomes a common assumption in many existing MSDA algorithms. **However, $\triangle$ in practice is implicit and always unobservable.** In this case, it would not be a good assumption to learn transferable features in DCTN.

4.3.2 Instance DA Loss in DCTN

Similar with Eq. 10, our target predictor in DCTN integrates M source-target relations with perplexity scores to reweight and aggregate the target category predictions from M category source classifiers. However, the multi-source perplexity scores are completely built on the discriminative results

[3] Since each sample x corresponds to an unique class y, $\{\mathscr{P}_j\}_{j=1}^M$ and $\mathscr{P}_t$ can be viewed as an equivalent embedding from $\{P_j(x, y)\}_{j=1}^N$ and $P_t(x, y)$ that we have discussed.

according to the multi-way adversarial learning principle instead of a presumed simplex in distribution weight combining rule. Although DCTN does not rely on the distribution weight combining rule, some meaningful upper bounds are also held to guarantee its target classification.

Specifically, according to Eqs. 2, 3, the cocktail target category predictor refers to a hypotheses h_t:

$$h_t(x) = \sum_{j=1}^{M} \frac{-\log(1 - D_j^*(x))}{\sum_{k=1}^{M} -\log(1 - D_k^*(x))} h_j(x), \tag{14}$$

where D_j^* ($\forall j \in [M]$) denotes the optimal domain discriminator with respect to the source j and the target.

Distinct from Eq. 10 where F is fixed, multi-way adversarial learning encourages DCTNs to learn domain-invariant features.

Lemma 1 $\forall j \in [M]$*, the optimal* D_j *corresponds to*

$$D_j^*(x) = \frac{\mathscr{P}_j(x)}{\mathscr{P}_j(x) + \mathscr{P}_t(x)}, \tag{15}$$

so that

$$h_t(x) = \sum_{j=1}^{M} \frac{\log\left(1 + \frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)}\right)}{\sum_{k=1}^{M} \log\left(1 + \frac{\mathscr{P}_k(x)}{\mathscr{P}_t(x)}\right)} h_j(x). \tag{16}$$

Note that h_t is proposed to classify target examples in the feature space, i.e., $x_t = F^*(\boldsymbol{x}_t)$, where F^* is the optimal feature extractor obtained by our multi-way adversarial learning, and $\boldsymbol{x}_t \sim \mathbf{X}_t$. Since target feature x_t is drawn from $\mathscr{P}_t$, it is reasonable to assume $\mathscr{P}_j(x_t) \leq \mathscr{P}_t(x_t)$ ($\mathscr{P}_j(x_t)$ denotes the probability of the target feature x_t drawn from the source j). Due to the adversarial DA manner, for $\forall j \in [M]$, $\mathscr{P}_j$ has been enforced to approach $\mathscr{P}_t$. Given this, it would be appropriate to suppose a source-specific approximation ratio $\alpha_j \in (0, 1)$ to describe the source-j-to-target adaptation, namely,

Assumption 1 (*Multi-way adversarial learning*) Provided a well-trained feature extractor F^*, $\forall j \in [M]$, it corresponds to an approximate ratio $\alpha_j \in (0, 1)$ so that $\forall x_t \in \mathscr{X}$, there exists $\alpha_j \mathscr{P}_t(x_t) \leq \mathscr{P}_j(x_t) \leq \mathscr{P}_t(x_t)$.

$\alpha_j \mathscr{P}_t(x_t) \leq \mathscr{P}_j(x_t)$ implies the upper bound of the discrepancy between $\mathscr{P}_t$ and $\mathscr{P}_j$ in the optimized feature space. α_j closer to 1 indicates the source j and the target more difficult to tell apart. x_t is drawn from a target domain rather than a source domain. To this end, it is more reasonable to assume $\mathscr{P}_j(x_t) \leq \mathscr{P}_t(x_t)$. Beyond this, we also consider the pseudo-labeling strategy in discriminative adaptation, leading to another assumption about pseudo-labeled examples:

Assumption 2 (*Pseudo-labeled discriminative adapatation*) Given a well-trained feature extractor and M source-specific classifiers, after each multi-way adversarial DA updates, target examples hold $\rho \in (0, 1)$ as a probability of false labels by the pseudo-labeling strategy.

which states $\rho \times 100\%$-at-least target examples whose categories are correctly forecast by our pseudo-annotating strategy. Based upon the assumptions, we develop an upper bound of a target classification error in terms of a given target feature x_t.

Proposition 2 *Suppose the converged feature extractor* F^* *satisfying Assumptions 1 and 2. Given a target feature* x_t*, its classification loss* $\mathscr{L}(\mathscr{P}_t, h_t, f)(x)$ *in DCTN can be upper bounded as follows:*

$$\mathscr{L}(\mathscr{P}_t, h_t, f)(x) \leq \frac{1}{\sum_{k=1}^{M} \log(1 + \alpha_k)} \sum_{j=1}^{M} \mathscr{P}_j(x) L\big(h_j(x), f(x)\big), \tag{17}$$

where $L\big(h_j(x), f(x)\big)$ *indicates an instance loss of the* j^{th} *source classifier.*

The target instance bound of DCTN is composed of a target feature's bounds on all source classifiers with its corresponding probabilities that it belongs to these sources. It indicates that, the closer a target feature is located at a source center, the higher its probability belongs to this source. So as long as there is a source suits well to a target feature (the adversarial learning and classification fine-tuning perform well in this source), the DCTN performance would be guaranteed since the classification bound and the probability that the target feature belongs to this source have jointly dominated the bound. It demonstrates the connection between DCTN and those methods based on multi-source mixture assumption to a target sample. However, DCTN do not rely on it since the probability that a target sample belongs to a source is based on their transferable features automatically obtained by adversarial learning instead of pre-given by human experience.

4.3.3 Vanilla MSDA

Based on the target instance loss with respect to h_t in Eq. 17, we provide the MSDA generalization bound of DCTN in the vanilla scenario:

Proposition 3 *Suppose the converged feature extractor* F^* *satisfying Assumptions 1 and 2. For all* $j \in [M]$*, the source maintains* $\mathscr{L}(\mathscr{P}_j, h_j, f) \leq \epsilon_j$, $(\epsilon_j > 0)$*. Then the expected loss of a mixture hypothesis* h_t *defined by Eq. 14 is at most* ϵ' *w.r.t. any target function* f : $\mathscr{L}(\mathscr{P}_t, h_t, f) \leq \epsilon'$*, where*

$$\epsilon' = \frac{1}{\sum_{k=1}^{M} \log(1+\alpha_k)} \left((1-\rho) \sum_{j\in[M]} \epsilon_j + M\rho \right) \tag{18}$$

Equation 18 denotes a surrogate target loss produced by the target category predictor [Eq.(2)], since it is not directly implemented to train the feature extractor F and M-source classifiers $\{C_j\}_{j=1}^{M}$. Equation 18 implies some guidances in transfer learning. Concretely, if it holds $\alpha_k \rightarrow 0, \forall k \in [M]$, then $\epsilon' \rightarrow 0$ and learning DCTN will fail. As long as some of source domains successfully approach the target ($\exists k \in [M]$, $\alpha_k \rightarrow 0$), ϵ' could provide a meaningful upper bound to reflect the MSDA process. Especially, when $\forall k \in [M]$ it holds $\alpha_k \rightarrow 1$, Eq. 18 would turn into a normal classification bound over the average of M-source classifiers. In terms of ρ, it shows the worst case about the mismatched categories in MSDA.

4.3.4 Category-Shift MSDAs

Though Eq. 18 is discussed in a vanilla MSDA scenario, the category predictor with source shifts also resembles the spirit. Concretely, we found that

$$\begin{aligned} C_t(c|\boldsymbol{x}^{(t)}) &= \sum_{c\in\mathscr{C}_j} \frac{s(\boldsymbol{x}^{(t)}; F, D_j)}{\sum_{c\in\mathscr{C}_k} s(\boldsymbol{x}^{(t)}; F, D_k)} C_j\big(c|F(\boldsymbol{x}^{(t)})\big) \\ &= \frac{\sum_{c\in\mathscr{C}_j} s(c|\boldsymbol{x}^{(t)}; F, D_j) C_j\big(c|F(\boldsymbol{x}^{(t)})\big)}{\sum_{c\in\mathscr{C}_k} s(c|\boldsymbol{x}^{(t)}; F, D_k) + \sum_{c\notin\mathscr{C}_k} s(c|\boldsymbol{x}^{(t)}; F, D_k)} \\ &\quad + \frac{\sum_{c\notin\mathscr{C}_j} s(c|\boldsymbol{x}^{(t)}; F, D_j) C_j\big(c|F(\boldsymbol{x}^{(t)})\big)}{\sum_{c\in\mathscr{C}_k} s(c|\boldsymbol{x}^{(t)}; F, D_k) + \sum_{c\notin\mathscr{C}_k} s(c|\boldsymbol{x}^{(t)}; F, D_k)} \\ &= \sum_{j\in[M]} \frac{s(c|\boldsymbol{x}^{(t)}; F, D_j) C_j\big(c|F(\boldsymbol{x}^{(t)})\big)}{\sum_{k\in[M]} s(c|\boldsymbol{x}^{(t)}; F, D_k)}. \end{aligned} \tag{19}$$

Given this, Eq. 3 could be viewed as the class-specific learner extended from Eq. 2. If $\{C_t(c|\boldsymbol{x}^{(t)}), c = 1, ..., M\}$ denotes a simplex with respect to all classes in the target domain, Eq. 3 turns into the special case of Eq. 2, thus, following the similar analysis.

In a MSDA problem with target category shifts, we conduct an upper bound of the target surrogate loss derived from Proposition 3. Let ρ' denote the proportion of target data wrongly labeled by our unknown-class discovery strategy,

Proposition 4 *Suppose the converged feature extractor F^* satisfying Assumptions 1 and 2. For all $j \in [M]$, the source maintains $\mathscr{L}(\mathscr{P}_j, h_j, f) \leq \epsilon_j$, $(\epsilon_j > 0)$. Then the expected loss of a mixture hypothesis h_t defined by Eq. 14 is at most ϵ' w.r.t. any target function $f : \mathscr{L}(\mathscr{P}_t, h_t, f) \leq \epsilon'$, where*

$$\epsilon' = \frac{(1-\rho')(1-\rho)\sum_{j\in[M]} \epsilon_j + \big((1-\rho')\rho + \rho'\big)M}{\sum_{k=1}^{M} \log(1+\alpha_k)}. \tag{20}$$

Equation 20 is upper bounded by Eq. 18. The equality is satisfied when $\rho' \rightarrow 0$, implying that no unknown target example has been missed to detect with our entropy-based "unknown" target example discovery strategy. In the source-target-category-shift scenario, learning DCTN could be considered as combining the analysis of the both category-shift scenarios.

5 Experiments

In the context of MSDA, we evaluate the classification accuracy of the target category predictor in experiments. Four adaptation learning cases, i.e., vanilla, source-category-shift, target-category-shift and source-target-category-shift MSDA problems, will be thoroughly studied. Each empirical study is implemented with a single GTX GeForce 1080 GPU on PyTorch platform. More implementation details are referred to the supplementary material.

5.1 Benchmarks and Measures

Four widely-applied DA benchmarks, i.e., *Office-31* (Saenko et al. 2010), *ImageCLEF-DA*, *Digits-five* and *DomainNet* (Peng et al. 2019) are introduced for the vanilla MSDA experimental evaluations. We follow the test routine in the previous works (Long et al. 2015, 2016) for fair comparisons. For reproducibility, the detailed dataset splits are released.[4]

– *Office-31* is a classical benchmark for object recognition with 31 categories. It has three datasets, **A** (*Amazon*), **D** (*DSLR*), **W** (*Webcam*). There are 4652 images in total.
– *ImageCLEF-DA* is released for the ImageCLEF 2014 domain adaptation challenge. It covers 12 object categories (aeroplane, bike, bird, boat, bottle, bus, car, dog, horse, monitor, motorbike, and people) shared in the three famous real-world datasets, **I** (*ImageNet ILSVRC 2012*), **P** (*Pascal VOC* 2012), **C** (*Caltech-256*). It includes 50 images in each class and totally 600 images for each domain.
– *Digits-five* includes five digit image sets drawn from following public datasets, **mt** (*MNIST*) (LeCun et al. 1998), **mm** (*MNIST-M*) (Ganin et al. 2017), **sv**(*SVHN*) (Netzer et al. 2011), **up** (*USPS*) and **sy** (*Synthetic Digits*) (Ganin et al. 2017), respectively. We draw 25,000 for training and 9,000 for testing in each set, i.e., *MNIST*, *MNIST-M*, *SVHN* and *Synthetic Digits* and choose the entire *USPS* dataset as one domain with only 9,298 images.
– *DomainNet* includes six natural image domain sets. *e.g.*, **clp**(*Clipart*), **inf**(*Infograph*), **pnt**(*Painting*), **qdr**(*Quickdraw*),

[4] http://www.sysu-hcp.net/deep-cocktail-network/.

rel(*Real*) and **skt**(*Sketch*), with 345 categories and around 0.6 million images in total.

Note that, DCTN's performance in the vanilla MSDA based on DomainNet have been provided in Peng et al. (2019). They used AlexNet as DCTN's backbone to compare with M3SDA in ResNet101. To this, we evaluate DCTN and M3SDA by using the same backbones in Office-31, ImageCLEF and Digits-five in the vanilla MSDA scenarios (Sect. 5.2). In terms of DomainNet, we standardized their backbones with ResNet101 and evaluated them when source and target category shifts both exist (Sect. 5.4).

As for the evaluation results, we follow the standard evaluation protocols adopted in unsupervised domain adaptation (Long et al. 2015; Ganin and Lempitsky 2015), and derive them to suit different MSDA scenarios (details are introduced in the corresponding sub-sections). Generally, for Office-31 and ImageCLEF-DA datasets, we use all labeled source examples and all unlabeled target examples. We compare the average classification accuracy of each method on three random independent experiments, and report the standard error of the classification accuracies by different experiments of the same transfer task. For the digit-5 and DomainNet benchmarks, we use all labeled source and unlabeled target training samples, then evaluate its performance on target test sets. We randomly run 3 times till the model converges and then choose the best results to report the accuracy. Finally, we perform model selection by tuning hyper-parameters using transfer cross-validation.

5.2 MSDA in Vanilla Scenarios

The existing work of MSDA lack comprehensive evaluations on complex real-world visual recognition. In our experiment, we introduce three traditional MSDA approaches, e.g., **RDALR** (Jhuo et al. 2013b), sparse FRAME (**sFRAME**) (Xie et al. 2015), **SGF** (Gopalan et al. 2011) as the baselines in *Office-31*, and two deep MSDA approaches Multi-Source Batch Normalization (**MSBN**) (Mancini et al. 2018) and **M3SDA** as the baselines in *Office-31* and *ImageCLEF-DA*. Besides, we also compare our DCTN with several single-source visual DA baselines, which include the conventional methods, e.g., Transfer Component Analysis (**TCA**) (Pan et al. 2011) and Geodesic Flow Kernel (**GFK**) (Gong et al. 2012), as well as several state-of-the-art deep DA approaches: Deep Domain Confusion (**DDC**) (Tzeng et al. 2015), Deep Reconstruction-classification Networks (**DRCN**) (Ghifary et al. 2016), Reversed Gradient (**RevGrad**) (Ganin and Lempitsky 2015), Pixel Domain Adaptation (**PixelDA**) (Bousmalis et al. 2017), Domain Adaptation Network (**DAN**) (Long et al. 2015), Residual Transfer Network (**RTN**) (Long et al. 2016) and Joint Adaptation Network (**JAN**) (Long et al. 2017). To achieve more comprehensive understanding about multi-source transfer, we compare our DCTN with these single source DA approaches by two different evaluation protocols. (1) *Single source*: Since they belong to single-source DA approaches, we directly report their single source transfer results from their original paper. (2) *Source combine*: multiple source domains are combined into a traditional single-source versus target domain adaptation setup. It helps to testify whether it would be able to boost the transfer performance gains through augmenting another source domain. Additionally, as baselines in the *Source combine* and multi-source standards, we use all images from sources to train backbone-based multi-source classifiers and apply them to classify target examples. These *Source only* results confirm whether our multi-source transfers are available (Negative indicates failure of adaptation). For fair comparisons, deep DA baselines in *Office-31* and *ImageCLEF-DA* employ the Alexnet backbones, and share the same backbone model (see our "Appendix") in *Digits-five*. The *Source-combine* results are basically derived from the official codes provided by their original papers.

Object recognition We report all transfer cases and compare our DCTN with the baselines in Tables 1 and 2 bolditalic, bold and italic indicate the performance of top 1, 2 and 3, respectively). Table 1 shows DCTN yielding the competitive results in the *Office-31* transfer tasks **A**,**W** → **D** and **A**,**D** → **W**, performing impressively in **D**,**W** → **A**. More specifically, DCTN significantly exceeds the traditional methods by a huge margin and mostly outperforms the single-source deep DA baselines, i.e., DAN, RTN, JAN, RevGred, and their source-combine variants. It reveals that if MSDA is treated as a single source DA problem by combining sources, the performance gain can not be fully excavated. Through the data transfer by DCTN, the potential power of multiple sources are efficiently used to boost the adaptation performance. Note that, MSBN is very competitive so that exceeds DCTN by 0.3% in Office-31 on its averaged accuracy. But MSBN does not generalize well across transfer cases: though achieving remarkable improvement in **D**,**W** → **A**, MSBN remains inconspicuous in **A**,**W** → **D** and **A**,**D** → **W** (fall behind source-combine single source DA variants). In a comparison, DCTN wins the top-3 performances in all transfer cases and thus, demonstrates more significant generalization ability. In *ImageCLEF-DA*, source-combine DA variants achieve more superior than their original single source models, whereas remains inferior to our DCTN. It validates that, no matter whether the domain size is equal or not, DCTN is able to learn more transferable and discriminative features than the other baselines, from multi-source transfer for natural image domains. MSBN completely fails in *ImageCLEF-DA* and even appears negative transfer compared with the source-only baseline.

Digit recognition Different from the previous visual recognition benchmarks, Digit-five contains five domains in

Table 1 Accuracy (%) on Office-31 in the vanilla MSDA setting

Standards	Models	W → D	A → D	A → W	D → W	D → A	W → A	Avg
Single source	Source only	99.0 ± 0.2	63.8 ± 0.5	61.6 ± 0.5	95.4 ± 0.3	51.1 ± 0.6	49.8 ± 0.4	70.1
	TCA	95.2 ± 0.0	60.8 ± 0.0	61.0 ± 0.0	93.2 ± 0.0	51.6 ± 0.0	50.9 ± 0.0	68.8
	GFK	95.0 ± 0.0	60.6 ± 0.0	60.4 ± 0.0	95.6 ± 0.0	52.4 ± 0.0	48.1 ± 0.0	68.7
	DDC	98.5 ± 0.4	64.4 ± 0.3	61.8 ± 0.4	95.0 ± 0.5	52.1 ± 0.6	52.2 ± 0.4	70.7
	DRCN	99.0 ± 0.2	66.8 ± 0.5	68.7 ± 0.3	96.4 ± 0.3	56.0 ± 0.5	54.9 ± 0.5	73.6
	RevGrad	99.2 ± 0.3	72.3 ± 0.3	73.0 ± 0.5	96.4 ± 0.3	53.4 ± 0.4	51.2 ± 0.4	74.3
	DAN	99.0 ± 0.3	67.0 ± 0.4	68.5 ± 0.5	96.0 ± 0.3	54.0 ± 0.5	53.1 ± 0.5	72.9
	RTN	99.6 ± 0.1	71.0 ± 0.2	73.3 ± 0.3	96.8 ± 0.2	50.5 ± 0.3	51.0 ± 0.1	73.7
	JAN	99.5 ± 0.1	71.8 ± 0.2	74.9 ± 0.3	96.6 ± 0.2	58.3 ± 0.3	55.0 ± 0.1	76.0
		A,W → D	A,D → W	D,W → A				
Source combine	Source only	98.1 ± 0.0	93.2 ± 0.0	50.2 ± 0.0				80.5
	RevGred	**99.0** ± 0.2	95.0 ± 0.4	55.1 ± 0.2				*83.0*
	DAN	98.4 ± 0.4	95.9 ± 0.3	53.6 ± 0.9				82.5
	RTN	*98.5* ± 0.4	***97.7*** ± 0.3	48.9 ± 0.9				81.7
	JAN	96.0 ± 0.4	94.0 ± 0.3	**57.2** ± 0.3				82.4
Multi-source	Source only	98.2 ± 0.0	92.7 ± 0.0	51.6 ± 0.0				80.8
	RDALR	31.2 ± 1.3	36.9 ± 1.1	20.9 ± 0.9				29.7
	sFRAME	54.5 ± 3.3	52.2 ± 1.4	32.1 ± 1.6				46.3
	SGF	39.0 ± 1.1	52.0 ± 2.5	28.0 ± 0.8				39.7
	MSBN	94.3 ± 0.4	94.0 ± 1.8	***61.5*** ± 1.5				**83.1**
	M3SDA	95.3 ± 0.4	*96.0* ± 0.8	35.5 ± 1.5				75.6
	DCTN (ours)	***100.0*** ± 0.0	**96.9** ± 0.1	*55.4* ± 0.2				***84.1***

Bolditalic, bold and italic indicate top 1, 2, 3 performances (Best viewed in bolditalic, bold and italic)

Table 2 Accuracy on ImageCLEF-DA in the vanilla MSDA setting

Standards	Models	I → P	C → P	I → C	P → C	P → I	C → I	Avg
Single source	Source only	66.2 ± 0.2	59.3 ± 0.5	84.3 ± 0.2	84.5 ± 0.3	70.0 ± 0.2	71.3 ± 0.4	73.9
	RevGrad	66.5 ± 0.5	63.5 ± 0.4	89.0 ± 0.5	88.7 ± 0.4	81.8 ± 0.4	79.8 ± 0.5	78.2
	DAN	67.3 ± 0.2	61.6 ± 0.3	87.7 ± 0.3	88.4 ± 0.2	80.5 ± 0.3	76.0 ± 0.3	76.9
	RTN	67.4 ± 0.3	62.0 ± 0.2	89.5 ± 0.4	90.1 ± 0.1	82.3 ± 0.3	78.0 ± 0.2	78.4
	JAN	67.2 ± 0.5	63.5 ± 0.4	**91.3** ± 0.3	91.0 ± 0.4	82.8 ± 0.4	80.0 ± 0.2	79.3
		I,C → P	I,P → C	P,C → I				
Source combine	Source only	68.3 ± 0.0	88.0 ± 0.0	81.2 ± 0.0				79.2
	RevGrad	66.7 ± 0.3	*90.2* ± 0.5	82.2 ± 0.1				79.7
	DAN	**69.1** ± 0.6	89.5 ± 0.4	81.3 ± 0.5				*79.9*
	RTN	65.3 ± 0.6	87.9 ± 0.4	80.0 ± 0.5				77.7
	JAN	*68.7* ± 0.2	89.4 ± 0.2	**82.6** ± 0.1				**80.2**
Multi-source	Source only	68.5 ± 0.0	89.3 ± 0.0	81.3 ± 0.0				79.7
	MSBN	64.4 ± 1.4	**90.3** ± 0.7	78.0 ± 0.5				78.1
	M3SDA	65.2 ± 2.1	87.6 ± 1.7	***83.8*** ± 0.5				78.7
	DCTN (ours)	***69.6*** ± 0.0	***91.0*** ± 0.1	**83.3** ± 0.2				***81.3***

Bolditalic, bold and italic indicate top 1,2,3 performances (Best viewed in bolditalic, bold and italic) total and is specified for multi-domain learning. We investigate 4-to-1 transfer results of DCTN within the following domain shifts: **mm, mt, sy, up** → **sv**; **mt, sv, sy, up** → **mm**; **mt, sv, mm, up** → **sy** and **mt, sv, sy, mm** → **up**, and provide the performance on average. We compare DCTN with RevGred, DAN and their source-combine transfer variants.

Overall accuracies of the baselines are concluded in Table 3. First of all, it is apparent that accuracies of single source DA approaches fall behind their source-combine. It implies that as M increases, multiple sources provide more evidences to boost transfer performance gains than those solely involved with a single source domain. However, we observe that these source-combine typically perform worse than their source-only except for **mt, sv, sy, up** → **mm**. In other words, despite of potential benefits multiple sources bring about, single source deep DA approaches conventionally suffer negative transfer. Therefore, it can not take advantage of the multi-source information into the model. In comparison, DCTN consistently shows positive transfer performances compared with the source only, and no matter of source-combine and multi-source ensemble, DCTN always outperforms the other baselines. In Table 4, the mean accuracy of our DCTN exceeds the second best by 3.6%.

5.3 MSDA in Source-Category-Shift Scenarios

In this subsection, we switch to evaluate DCTN in the category shift scenario, where the multiple sources do not share the same categories. We compare our DCTN with state-of-the-art approaches, i.e., DAN and RevGred, under the single-source and source-combine evaluation settings. Our experiments are conducted in four MSDA transfer cases: **A**,**D** → **W** and **A**,**W** → **D** in *Office-31*; **I**,**P** → **C** and **C**,**P** → **I**.

Evaluation protocol Since source-category-shift is newly proposed in MSDA scenario, benchmarks should be amended to to evaluate DA algorithms in this scenario. Specifically, suppose that M sources involve C categories and $C_p \leq C$ indicates the number of their public classes. Due to $M = 2$, we consider the alphabetical order of the C classes and take the first $\frac{C-C_p}{2}$ and last $\frac{C-C_p}{2}$ classes as the source-specific private classes, then the rest proportion $\frac{C_p}{C}$ denotes the public classes. To unveil the comprehensive the baselines in this scenario, we evaluate them by specifying the public-class proportions $\frac{C_p}{C}$ in $\{0, 0.3, 0.5, 0.7, 1\}$, respectively.

We elaborate three metrics to reflect the adaptation capability of baselines from different perspectives. First, classification accuracy is to evaluate whether the baseline helps the classifier address the domain/category shift problem.

Second, We employ a relative measure termed *degraded accuracy* by examining how much performance drops when source-category shift exists, which is simply calculated as follows:

$$DA\left(\frac{C_p}{C}\right) = Acc\left(\frac{C'_p}{C} = 1\right) - Acc\left(\frac{C_p}{C}\right), \tag{21}$$

where $Acc(\frac{C_p}{C})$ denotes the accuracy when the public-class proportion is $\frac{C_p}{C}$, and $Acc(\frac{C'_p}{C} = 1)$ means the accuracy of the model trained in vanilla MSDA scenario. The formula showcases the performance drop caused by inconsistent categories of sources. The lower value means the algorithms less affected by this negative effect, performing more robust in this scenario. Finally, we employ *transfer gain* as the third metric to further confirm the availability of transfer learning. Transfer gain is calculated through subtracting the baseline's accuracy with the accuracy of Source only. A positive value undoubtedly means that the transfer is available, while a negative value means the DA approach aggravates the domain shift problem.

Results The experiments cover the four transfer cases. Experimental results on these metrics (mean accuracy, degraded accuracy and transfer gain), are illustrated in Figs. 3, 4 and 5, respectively. DCTN always outperforms other baselines in different proportions of public classes and transfer cases. Generally, the improvement becomes larger as the sources contain more public classes. In Fig. 4, it can be observed that both Source only and DCTN behave neck and neck in *ImageCELF-DA*.

Considering the relative enhancement measure, in *Office-31*, Source-only even obtains lower DA values than DCTN. Note that, it does not imply Source-only outperforms our DCTN. In particular, compared with DA algorithms, i.e., DAN, RevGred and DCTN, Source-only undergoes fully-supervised learning, therefore, is free of the risk caused by category misalignment. To some extent, it should be treated as a sort of consecutive strategy preferring the safety of supervised training rather than adapting to a domain without labeled data.

Considering the absolute performance shown in Fig. 3, it is obvious that, Source-only are almost inferior to all DA approaches.

Besides of superior transfer performance improvement, another merit of DCTN is the strong resistance against the potential negative transfer influences. As demonstrated in Fig. 5, compared with other state of the art methods, DCTN remain positive values in all transfer cases. Specifically, in *Office-31*, DAN shows impressive transfer performance in **A**,**D** → **W** with 0% public classes, but its performance on transfer gain is quite unstable as the public class number becomes challenging. RevGred performs more stable and better than DAN in general, whereas both of them inevitably suffer from negative transfer and are wholly suppressed by our DCTN. Similarly, in *ImageCLEF-DA*, DAN

Table 3 Classification accuracy (%) on Digits-five dataset for MSDA in the vanilla setting

Standards	Models	mm → sv	mt → sv	sy → sv	up → sv	mt → mm	sv → mm	sy → mm	up → mm
Single source	Source only	45.3	46.4	67.4	29.7	58.0	49.6	54.8	43.7
	RevGred	49.2	38.8	66.9	25.8	83.6	45.1	50.6	40.0
	DAN	43.2	42.2	67.1	38.5	53.5	51.8	58.8	40.5
	PixelDA	29.2	28.9	84.2	26.2	86.3	11.6	26.7	29.1

		mm, mt, sy, up → sv	mt, sv, sy, up → mm
Source combine	Source only	*74.4*	65.5
	RevGred	68.9	***71.6***
	DAN	71.0	66.6
	PixelDA	62.3	61.6
Multi-source	Source only	64.6	60.7
	MSBN	66.4	63.1
	M3SDA	**78.6**	*69.7*
	DCTN (ours)	***78.7***	***71.6***

Standards	Models	mm → sy	mt → sy	sv → sy	up → sy	mt → up	sv → up	sy → up	mm → up
Single source	Source only	34.7	39.6	90.7	27.1	77.6	64.8	81.5	51.1
	RevGred	55.4	55.8	88.3	43.0	90.9	76.3	84.9	80.3
	DAN	40.6	32.1	84.0	29.1	87.6	57.4	80.6	65.0
	PixelDA	43.6	43.2	10.7	32.9	59.5	13.4	20.7	25.2

		mm, mt, sv, up → sy	mt, sv, sy, mm → up
Source combine	Source only	**92.3**	89.4
	RevGred	90.8	88.7
	DAN	*91.3*	90.0
	PixelDA	60.5	19.4
Multi-source	Source only	90.0	86.1
	MSBN	88.4	***95.4***
	M3SDA	87.6	**95.2**
	DCTN (ours)	***93.1***	*91.7*

Table 4 Average accuracy (%) performances of the above baselines

Single source			Single combine				Multi-source				
Source only	RevGred	DAN	PixelDA	Source only	RevGred	DAN	PixelDA	Source only	MSBN	M3SDA	DCTN(ours)
53.9	60.9	54.5	35.7	*80.4*	80.0	79.7	51.0	75.4	78.3	**82.7**	***84.0***

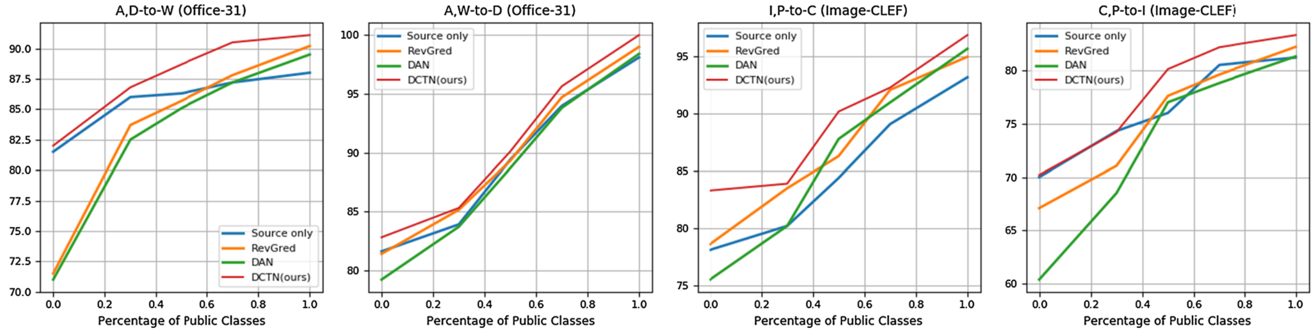

Fig. 3 The absolute performances based upon the mean accuracies (%) of Source only, RevGread, DAN and DCTN on Office-31 and ImageCLEF-DA under the MSDA category shift scenario. The curves denote their accuracies changing as the public classes across multiple sources increase. Higher is better

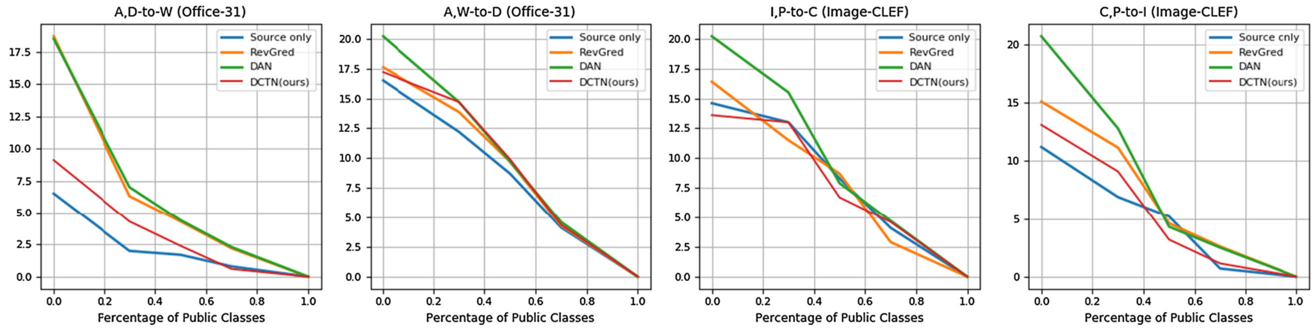

Fig. 4 The relative performance (degraded accuraies, the accuracy under vanilla scenario minus the the accuracy under category shift) of Source only, RevGread, DAN and DCTN on Office-31 and ImageCLEF-DA under MSDA category shift scenario. The curves denote how much their accuracies drop as the public classes across multiple sources increase. Lower is better

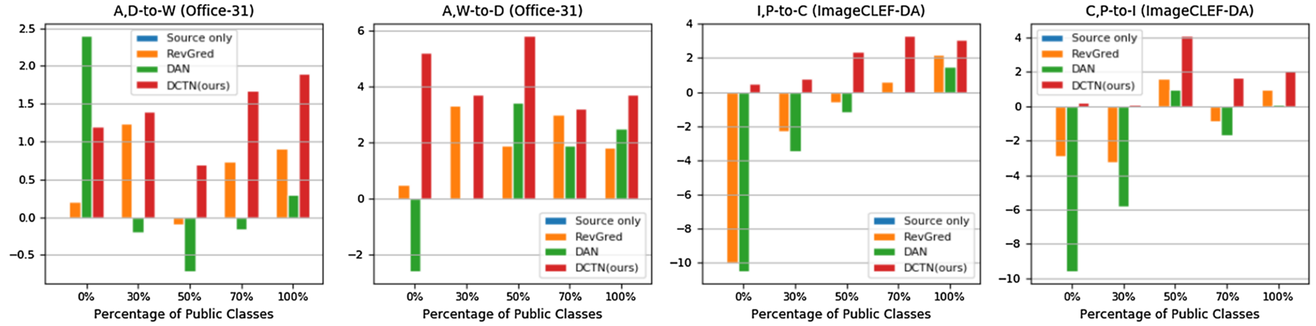

Fig. 5 The transfer gains (the accuracy of the baseline minus the accuracy of source only) of Source only, RevGread, DAN and DCTN on Office-31 and ImageCLEF-DA under MSDA category shift scenario. The negative value means the negative transfer, which causes even heavier model damage than those without domain adaptation. Higher is better

Table 5 Accuracy (%) of each method based on the 10-shared-class target-category-shift scenario in Office-31

	Evaluation Protocols	W → D		A → D		A → W		D → W		D → A		W → A		Avg	
		OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
Single source	OSVM	62.5	*59.2*	59.6	*59.1*	57.1	*55.0*	44.1	*39.3*	14.3	*5.9*	13.0	*4.5*	40.6	*37.1*
	MMD+OSVM	62.0	58.5	47.8	44.3	41.5	36.2	34.4	28.4	9.9	0.9	11.5	2.7	34.5	28.5
	BP+OSVM	49.7	44.8	40.3	35.6	31.0	24.3	33.6	27.3	10.4	1.5	11.5	2.7	29.5	22.7
	ATI-λ+OSVM	*92.7*	–	*72.0*	–	*65.3*	–	*82.2*	–	66.4	–	*71.6*	–	*75.0*	–
	RevGred-OP	**96.8**	**96.9**	**76.6**	**76.4**	**70.1**	**69.1**	**94.4**	**94.6**	*62.5*	**62.3**	**82.3**	**82.2**	**80.4**	**80.2**
		A,W → D				A,D → W				D,W → A					
	Evaluation Protocols	OS		OS*		OS		OS*		OS		OS*		OS	OS*
	DCTN (ours)	***97.1***		***99.4***		***96.7***		***98.9***		***78.8***		***73.7***		***90.9***	***90.7***

Best viewed in bolditalic, bold, italic

and RevGred still fail to achieve a promising transfer performance. In particular, when the number of public classes is small, their transfers even result in more model damages.

5.4 MSDA in Target-Category-Shift Scenarios

In this subsection, we evaluate DCTN in the target-category-shift scenario. As we previously discussed, it can be viewed as an open-set DA problem in a multi-source condition. We follows the similar experimental setting by reconfiguring *Office-31* benchmark as Saito et al. (2018). Concretely, we randomly choose the 10 classes in the *Caltech* dataset (Gong et al. 2012) as the common classes of the sources and target and the rest 21 are "unknown". In order to fairly compare with the single-domain open-set DA methods (Saito et al. 2018; Busto and Gall 2017) (they can be treated as Source-combine baselines in the target-shift experiments), we follow their protocols. Specifically, we evaluate all baselines on three domains in Office-31 with different numbers and for each domain, 1–11 classes are selected as shared classes across sources and target; 21–31 classes are selected as unknown target classes for identification. We accept the routine adapted in Saito et al. (2018) so that 11–20 classes have been abandoned.

Baselines For a fair comparison, we compare five state-of-the-art open-set DA approaches with our DCTN in target-category-shift MSDA scenario: **OSVM**, **MMD + OSVM**, **BP + OSVM**, **ATI-λ + OSVM**, and **RevGred-OP** (Saito et al. 2018). The first four methods are derived from Open-set SVM (**OSVM**) (Busto and Gall 2017), which employ a threshold to preclude the target examples probably belonging to the "unknown" class. The last one is developed from RevGred. Since they are not open-sourced, to ensure the fairness in our comparisons, we directly report their published performance results in the single source open-set scenario.

Evaluations Two evaluation measures, i.e., OS and OS*, are used to evaluate DCTN and comparison methods. The first testifies the methods on all target categories. The second evaluates them on 10 known categories. As can be observed in Table 5, in evaluation criteria OS and OS*, the single best accuracies of RevGred-OP remain suppressed by DCTN in **A**,**W** → **D** (97.1, 99.4 of DCTN better than 96.8, 96.9 of RevGred-OP) and **A**,**D** → **W** (96.7, 98.9 of DCTN better than 94.4, 94.6 of RevGred-OP).

5.5 MSDA in Source-Target-Category-Shift Scenarios

Source and target category shifts may concurrently appear. How much the joint negative transfer they bring about and whether it can be mitigated by MSDA algorithms, remain underexplored. To this, we start the evaluation from the experimental setup in MSDA with the target category shift, then further vary the proportion of the public categories, similar to the source-shift practice in Section 5.3. Our experiments are conducted in four transfer cases: **A**,**D** → **W** and **A**,**W** → **D** in Office-31; **clp**,**pnt**,**dqr**,**rel**,**skt** → **inf** and **clp**,**pnt**,**dqr**,**rel**,**inf** → **skr** in DomainNet. For DomainNet, we selected the categories with IDs over 250 and unify them to construct the unknown class c_u. Afterwards, we change the proportions of public categories from 0%, 50% to 100%. Since DomainNet is a very challenging benchmark. So we slightly change the data-split routine in Office-31. Specifically, in the case of 0%, we select 1–125 classes into the first and second sources, 126–250 classes into the third and fourth sources, then let the last source contains all 250 categories; in the case of 50%, we select 1–166 categories into the first and second sources, 84–250 categoies into the third and fourth sources, the last source contains all 250 categories. So the public classes in DomainNet refers to those shared across source 1,2 and source 3,4. This setting simplifies the complex category relation across the five source domains, encourage the evaluation to focus on the varation of performances across baselines.

In terms of the baselines, distinct from what were evaluated in the previous sub-section, we considered the comparison between M3SDA and DCTN along with Source-only.

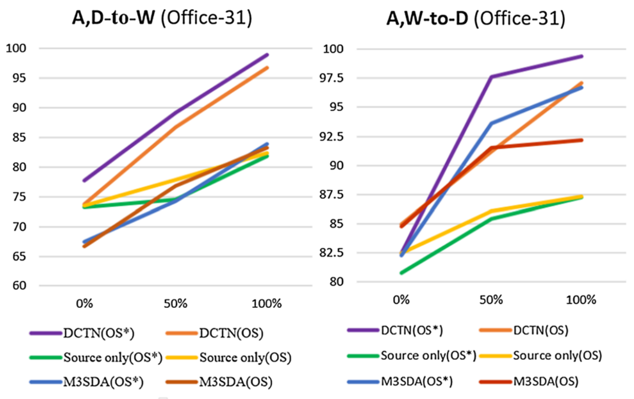

Fig. 6 The accuracies (%) of Source only, M3SDA and DCTN based on two transfer cases, in the source-target-category-shift scenario on Office-31 (OS and OS* indicate two evaluation protocols in target category shift scenarios). The curves denote their accuracies changing as the percentage of public classes across multiple sources increases. Higher is better

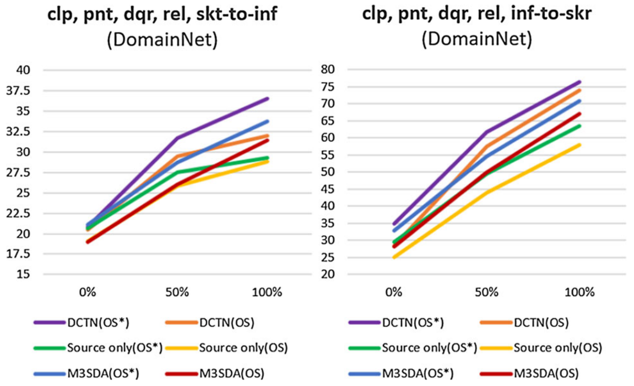

Fig. 7 The accuracies (%) of Source only, M3SDA and DCTN based on two transfer cases, in the source-target-category-shift scenario on DomainNet (OS and OS* indicate two evaluation protocols in target category shift scenarios). The curves denote their accuracies changing as the percentage of classes across multiple sources increases. Higher is better

Since M3SDA and DCTN are both state-of-the-art MSDA algorithms while with a similar spirit behind, thus, it would be insightful whether DCTN and M3SDA can both prevent the negative transfer or not. Since the original M3SDA algorithm is unable to handle the unknown categories in the target domain, for a fair comparison, we endowed M3SDA with the identical strategy in DCTN to screen the unknown-class samples. All baselines are evaluated based on the classification accuracy under OS and OS* criterion.

The results are illustrated in Figs. 6 and 7. In **A**,**W** → **D** , **clp**,**pnt**,**dqr**,**rel**,**inf** → **skr** and **clp**,**pnt**,**dqr**,**rel**,**skr** → **inf**,all the accuracy curves of DCTN and MSDA performed as upper envelopes of the Source-only, showing that the negative transfer effects have been eliminated in the cases. Notably, DCTN keeps ahead in all cases and protocols. But when the proportion of public classes decreases, the transfer gains brought by DCTN and MSDA are gradually minimized. Especially, when there are no public categories across the source domains, the transfer improvement from M3SDA has been completely erased in all transfer cases. In **A**,**D** → **W**,M3SDA has suffered a serious negative transfer effects in the OS and OS* protocols. Instead, DCTN is still able to provide a transfer gain in this case. The results showcase the superiority of DCTN in these tough category-shift scenario.

5.6 Model Analysis

Feature visualization Take the task of **A**,**D** → **W** in *Office-31* for example.

We visualize the DCTN activations before and after adaptation, which is impressive to demonstrate that the transferability learnt by DCTN. For simplicity, both source domains have been separated to emphasize the contrast of the target domain. As we can see in Fig. 8, compared with those from source only, our activations from **A** → **W** and **D** → **W** have shown good adaptation patterns. This indicates DCTN can successfully learn transferable features with multiple sources. Besides, the target activations become more clear to categorize, which suggests that the features learned by DCTN attains desirable discriminative property. Finally, even if the multi-source transfer has been composed of hard transfer task (**A** → **W**), DCTN is still able to adapt to target domain without the performance degradation in **D** → **W**.

Ablation study The learning of DCTN consists of the multi-way adversary and auto-labeling scheme. To further reveal their function, we decompose DCTN into two variants: The *adversarial-only* model excludes the pseudo-labels and updates the category classifier with source samples. The *pseudo-only* model forbids the adversary and categorize target samples with average multi-source results. As shown in Fig. 9a, the accuracy of adversary behaves stably in each iteration. But due to the lack of target class guidance, its final performance hits a bottleneck. Without the adversary, the accuracy of pseudo labels significantly drops and pulls down the DCTN accuracy. It indicates that both adaptations cooperate with each other to achieve desirable transfer behaviors. Diving deeper in Fig. 9b, the test accuracy and the pseudo label accuracy show converged in the alternative learning, which implicitly reveals the consistency between both adaptations. We also provide the ablation to the domain batch mining technique (Table 6), which testifies the method's efficacy.

Pseudo-labeling strategy From the ablation above, we know pseudo-labeled target samples are playing a key role of training a well-performed DCTN. So it is important to check whether annotation strategy may improve other baseline. To this, we evaluate DAN, RTN, JAN, RevGred (four single-source DA algorithms with their source-combine results),

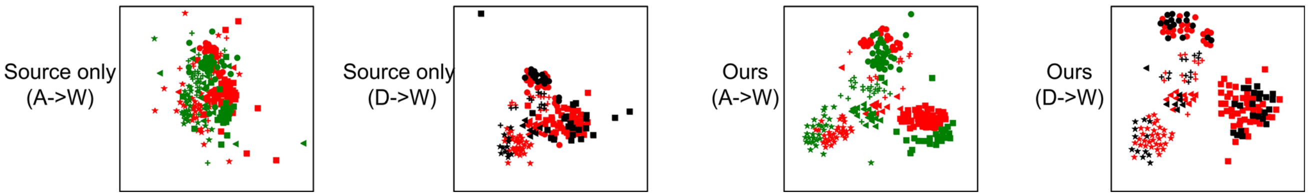

Fig. 8 The t-SNE (Maaten and Hinton 2008) visulization of A,D → W. Green, black and red represent domains A, D and W respectively. We use different markers to denote 5 categories, e.g., bookcase, calculator, monitor, printer, ruler. Best viewed in color

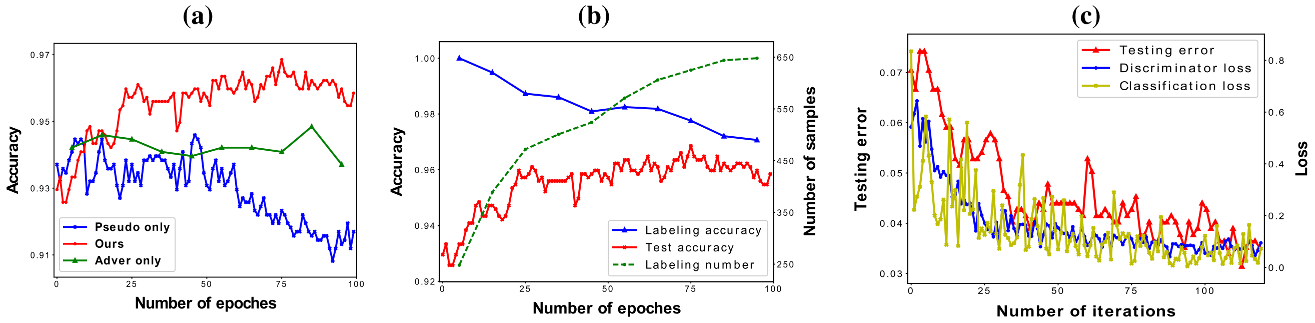

Fig. 9 Analysis: **a** the accuracies of DCTN, adversarial-only and pseudo-only models; **b** the accuracies of testing samples and pseudo-labeled target samples; **c** the convergence performance on different losses. Best viewed in color

Table 6 Ablation study of Algorithm 1 in Office-31

	A,W → D	A,D → W	D,W → A	Overlap	Disjoint
w	**100.0** ± 0.0	**96.9** ± 0.1	**55.4** ± 0.2	**90.2** ± 0.1	**82.9** ± 0.2
w/o	99.1 ± 0.3	96.3 ± 0.2	55.2 ± 0.2	89.4 ± 0.2	82.6 ± 0.1

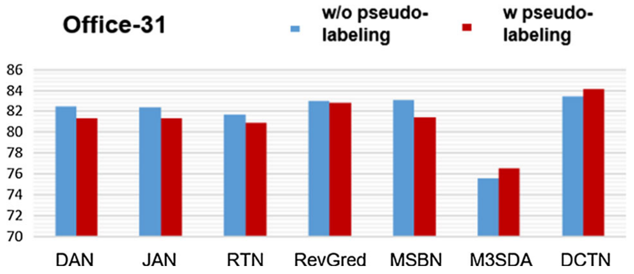

Fig. 10 The comparison of different algorithms when they use and don’t use pseudo-labeled target samples in training

MSBN, M3SDA and our DCTN (three MSDA algorithms) independently when they are learned with and without using our pseudo-labeling strategy. In Fig. 10, we observed that the accuracies of all single-source approaches and MSBN have been decreased, probably due to using a single classifier for prediction. Instead, M3SDA and DCTN are benefited from the pseudo-labeling strategy.

Convergence analysis As DCTN involves a complex learning procedure including adversarial learning and alternative adaptation, we testify the convergence performance of different losses. In the process of hard transfer A → W, Fig. 9c demonstrates that, despite deviation, the classification loss, adversarial loss and test error gradually converge.

6 Conclusion

In this paper, we have explored the unsupervised DA involved with multiple sources challenged by domain shift and category shift. Beside the vanilla MSDA transfer scenario, we further investigate three other innovative and realistic MSDA scenarios, where the category sets across multiple sources and the target are assumed to be inconsistent. In order to overcome these transfer challenges, we propose *deep cocktail network* (DCTN), an adversarial DA framework to obtain transferable and discriminative features from multiple sources to a target domain. It constitutes an alternating learning process that delicately refers to our target classification principle. DCTN can be flexibly deployed in ordinary MSDA and category shift scenarios, and more importantly, it suits the open-set scenario with a mild reconfiguration. Delving into the motivation of DCTN, we further reveal that, DCTN connects with a previous MSDA theory and enjoys an expected loss upper bound through an adversarial DA assumption instead of specifying a strong target mixture precondition. Finally, DCTN is evaluated across three benchmarks with massive transfer combinations under three scenarios. It achieves state-of-the-art results in most of our evaluation criteria and behaves extraordinarily to resist negative transfer effects.

Acknowledgements This work was supported in part by NSFC (Nos. 62006253, U181146, 61836012, 61976233), State Key Development Program (No. 2018YFC0830103), Fundamental Research Funds for the Central Universities (No. 19lgpy228), and Major Project of Guangzhou Science and Technology of Collaborative Innovation and Industry under Grant 201605122151511. We also thank Ruijia Xu for his valuable suggestion to the revision.

Appendix A: Proofs

Proof (Proof (**Lemma** 1)) Suppose the optimal feature extractor is F^{*}, then M source-target adversarial learning pairs can be separately considered and correspond to the optimization objectives w.r.t. $\{D_j\}_{j=1}^{M}$, respectively. Substitute $F^{*}(\boldsymbol{x})$ by x, and there is

$$D_j^{*}(x) = \frac{\mathscr{P}_j(x)}{\mathscr{P}_j(x) + \mathscr{P}_t(x)} \tag{22}$$

is exactly derived from Theorem.1 in Goodfellow et al. (2014).

$$\begin{aligned} h_t(x) &= \sum_{j=1}^{M} \frac{-\log\left(\frac{\mathscr{P}_t(x)}{\mathscr{P}_j(x)+\mathscr{P}_t(x)}\right) h_j(x)}{\sum_{k=1}^{M} -\log\left(\frac{\mathscr{P}_t(x)}{\mathscr{P}_j(x)+\mathscr{P}_t(x)}\right)} \\ &= \sum_{j=1}^{M} \frac{\log\left(1+\frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)}\right) h_j(x)}{\sum_{k=1}^{M} \log\left(1+\frac{\mathscr{P}_k(x)}{\mathscr{P}_t(x)}\right)}. \end{aligned} \tag{23}$$

□

Proof (Proof (**Proposition** 2)) Given Lemma 1, it holds

$$\begin{aligned} h_t(x) &= \sum_{j=1}^{M} \frac{\log\left(1+\frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)}\right)}{\sum_{k=1}^{M} \log\left(1+\frac{\mathscr{P}_k(x)}{\mathscr{P}_t(x)}\right)} h_j(x) \\ &\leq \sum_{j=1}^{M} \frac{\log\left(1+\frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)}\right)\, h_j(x)}{\sum_{k=1}^{M} \log(1+\alpha_k)} \\ &\leq \sum_{j=1}^{M} \frac{\mathscr{P}_k(x)\, h_j(x)}{\mathscr{P}_t(x) \sum_{k=1}^{M} \log(1+\alpha_k)}, \end{aligned} \tag{24}$$

To this end, given a target feature x,

$$\begin{aligned} &\mathscr{L}(\mathscr{P}_t, h_t, f)(x) \\ &= L\left(\sum_{j=1}^{M} \frac{\log(1+\frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)})}{\sum_{k=1}^{M} \log\left(1+\frac{\mathscr{P}_k(x)}{\mathscr{P}_t(x)}\right)} h_j(x), f(x)\right) \mathscr{P}_t(x) \\ &\leq \sum_{j=1}^{M} \frac{\log\left(1+\frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x)}\right)}{\sum_{k=1}^{M} \log(1+\frac{\mathscr{P}_k(x)}{\mathscr{P}_t(x)})} \mathscr{P}_t(x) L\left(h_j(x), f(x)\right) \\ &\leq \sum_{j=1}^{M} \frac{\mathscr{P}_j(x)}{\mathscr{P}_t(x) \sum_{k=1}^{M} \log(1+\alpha_k)} \mathscr{P}_t(x) L\left(h_j(x), f(x)\right) \\ &= \sum_{j=1}^{M} \frac{\mathscr{P}_j(x)}{\sum_{k=1}^{M} \log(1+\alpha_k)} L\left(h_j(x), f(x)\right) \end{aligned} \tag{25}$$

in which the first inequality is derived from the convexity of the loss function $L(\cdot, \cdot)$. □

Proof (**Proposition** 3) In terms of Proposition 1, we provide the upper bound of $\mathscr{L}(\mathscr{P}_t, h_t, f)$. Specifically,

$$\begin{aligned} \mathscr{L}(\mathscr{P}_t, h_t, f) &= \int_{x\in\mathscr{X}} \mathscr{L}(\mathscr{P}_t, h_t, f)(x)dx \\ &\leq \int_{x\in\mathscr{X}} \sum_{j=1}^{M} \frac{\mathscr{P}_j(x)}{\sum_{k=1}^{M} \log(1+\alpha_k)} L\big(h_j(x), f(x)\big)dx \\ &= \frac{1}{\sum_{k=1}^{M} \log(1+\alpha_k)} \sum_{j=1}^{M} \\ &\quad \int_{x\in\mathscr{X}} \mathscr{P}_j(x) L\big(h_j(x), f(x)\big)dx \end{aligned} \tag{26}$$

Since ρ indicates the proportion of wrongly-labeled target data by the auto-annotating strategy in the discriminative adaptation phase; $f'(x)$ represents the wrong target function w.r.t. x, namely, $\forall x \in \mathscr{X}$, it holds $f'(x) \neq f(x)$ and $L\big(h_j(x), f(x)\big) \leq L\big(h_j(x), f'(x)\big)$. Therefore,

$$\begin{aligned} \mathscr{L}(\mathscr{P}_t, h_t, f) &\leq \frac{1}{\sum_{k=1}^{M} \log(1+\alpha_k)} \\ &\quad \sum_{j=1}^{M} \int_{x\in\mathscr{X}} L\big(h_j(x), f(x)\big)dx \\ &= \frac{1}{\sum_{k=1}^{M} \log(1+\alpha_k)} \\ &\quad \sum_{j=1}^{M} \Bigg(\int_{x\in\mathscr{X}} \mathscr{P}_j(x)\Big((1-\rho)L\big(h_j(x), f(x)\big) \\ &\quad + \rho L\big(h_j(x), f(x)\big)\Big)dx\Bigg), \\ &\leq \frac{1}{\sum_{k=1}^{M} \log(1+\alpha_k)} \\ &\quad \sum_{j=1}^{M} \Bigg(\int_{x\in\mathscr{X}} \mathscr{P}_j(x)\Big((1-\rho)L\big(h_j(x), f(x)\big) \\ &\quad + \rho L\big(h_j(x), f'(x)\big)\Big)dx\Bigg), \end{aligned} \tag{27}$$

Due to the assumption of the 0-1 loss function on L, it follows the analysis in Saito et al. (2017) and holds

$$\begin{aligned}\int_{x\in\mathscr{X}} \mathscr{P}_j(x)L\big(h_j(x), f(x)\big)dx &\le \int_{x\in\mathscr{X}} \mathscr{P}_j(x)L\big(h_j(x), f'(x)\big)dx \\ &\le \int_{x\in\mathscr{X}} \mathscr{P}_j(x)dx = 1.\end{aligned} \tag{28}$$

so that

$$\mathscr{L}(\mathscr{P}_t, h_t, f) \le \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\Big((1-\rho)\sum_{j\in[M]}\epsilon_j + M\rho\Big) \tag{29}$$

Conclude the proof. □

Proof (**Proposition 4**) According to the results of Proposition.21, we have

$$\mathscr{L}(\mathscr{P}_t, h_t, f)(x) \le \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\sum_{j=1}^{M}\mathscr{P}_j(x)L\big(h_j(x), f(x)\big).$$

Therefore

$$\begin{aligned}\mathscr{L}(\mathscr{P}_t, h_t, f) &= \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\int_{x\in\mathscr{X}}\sum_{j=1}^{M}\mathscr{P}_j(x)L\big(h_j(x), f(x)\big)(x)dx \\ &\le \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\sum_{j\in[M]}\Big((1-\rho')\int_{x\in\mathscr{X}}\mathscr{P}_j(x)L\big(h_j(x), f(x)\big)dx \\ &\quad \int_{x\in\mathscr{X}}\mathscr{P}_j(x)L\big(h_j(x), f'(x)\big)dx\Big) \\ &\le \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\sum_{j\in[M]}\Big((1-\rho')\Big((1-\rho)\int_{x\in\mathscr{X}}\mathscr{P}_t(x)L\big(h_j(x), f(x)\big)dx \\ &\quad + \rho\int_{x\in\mathscr{X}}\mathscr{P}_t(x)L\big(h_j(x), f'(x)\big)dx\Big) \\ &\quad + \rho'\int_{x\in\mathscr{X}}\mathscr{P}_t(x)L\big(h_j(x), f'(x)\big)dx\Big),\end{aligned} \tag{30}$$

where the first, third and fifth inequalities are derived from the proof of Proposition 2; the second inequality is developed from the entropy-based unknown category discovery strategy, which is specified in target-category-shift and source-target-category-shift scenarios (Notice that, since the unknown class discovery is executed ahead of the pseudo-labeling strategy, it makes the the inequality w.r.t. ρ nested in the inequality w.r.t. ρ'); the fourth inequality is derived from the 0–1 loss function upper bound discussed in Saito et al. (2017) .

Due to the assumption of the 0–1 loss function on L, it follows (Saito et al. 2017) and holds

$$\begin{aligned}\mathscr{L}(\mathscr{P}_t, h_t, f) &\le \sum_{j\in[M]}\frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\Big((1-\rho')(1-\rho) \\ &\quad \int_{x\in\mathscr{X}}\mathscr{P}_t(x)L\big(h_j(x), f(x)\big)dx + \rho(1-\rho') + \rho'\Big) \\ &\le \frac{1}{\sum_{k=1}^{M}\log(1+\alpha_k)}\Big((1-\rho')(1-\rho)\sum_{j\in[M]}\epsilon_j \\ &\quad + \big((1-\rho')\rho + \rho'\big)M\Big)\end{aligned} \tag{31}$$

□

Appendix B: Implementation details

The setup of γ and ζ The the pseudo labeling strategies of DCTN rely on hyper-parameters γ and ζ. The threshold γ is leveraged to select a part of target candidates, which are annotated as “high-confident” and augmented with multi-source examples to train the multi-source classifiers. We set the value over 90% to ensure the quality of selecte target samples. Instead of choosing a specific threshold γ, we rank the target examples according to their entropy values on the source-specific classifiers by a monotonically decreasing order, then choose the top 15% as the “unknown” candidates: 300/120/140 as the “unknown” candidates in A/D/W domains, respectively. This manner promises adequate “unknown” examples to train a reliable classifier for each source domain. The schemes is adopted the same in the experiment of DomainNet (Details see Table 7).

Network Implementation details For the recognitions in Office-31 and ImageCLEF-DA, existing deep DA approaches (Long et al. 2015, 2016) routinely employ Alexnet (Krizhevsky et al. 2012) as their backbones. For a fair comparison, we choose a DCTN architecture deriving from the Alexnet pipeline. As Fig. 11 illustrated, the representation module F is designed as a five-layer fully-convolutional network with three max-pooling operators, and the (multi-source) category classifier C is a three-layer fully-connected multi-task network. They are stacked into an exactly Alexnet-like pipeline to categorize examples. We adopt a CNN with a two-head classifier as domain discriminator D.

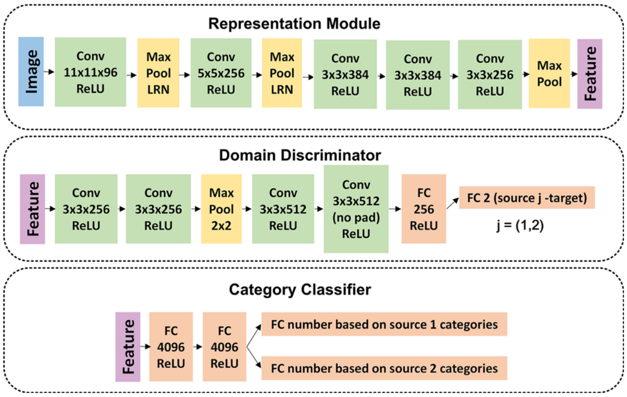

Fig. 11 The representation module, domain discriminator and category classifier we used in the experiments about object recognition. (Best viewed in color)

For the sake of legibility, we apply the sigmoid cross entropy loss to denote the multi-way adversarial learning inducing the perplexity score in our paper. Under M adversarial adaptation context, this loss function leads to the gradient vanishing and behaves extremely unstable during training. To overcome this issue, we replace it with the least square measure (Mao et al. 2017) in practice to ensure robust adversarial learning:

$$\mathscr{L}_{adv}^{(ls)}(F, D) = \frac{1}{M}\sum_{j}^{M}\mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_j}[(D_j(F(\boldsymbol{x})))^2] + \mathbb{E}_{\boldsymbol{x}^{(t)}\sim\mathbf{X}_t}[(1 - D_j(F(\boldsymbol{x}^{(t)})))^2]. \tag{32}$$

Accordingly, the confusion loss has been revised as

$$\mathscr{L}_{cf}^{(ls)}(\boldsymbol{x}; F, D_j) = \left(D_j(F(\boldsymbol{x})) - \frac{1}{2}\right)^2. \tag{33}$$

Then given a target instance $\boldsymbol{x}^{(t)}$, a least square perplexity score is

$$s(\boldsymbol{x}^{(t)}; F, D_j) = (D_j(F(\boldsymbol{x}^{(t)})))^2. \tag{34}$$

The implementation keeps consistent with all our analysis in the paper. No matter in training or test, we need a perplexity score weighting scheme to predict the class of the target instance. While in the adversarial learning process, the domain discriminator D must be gradually trained to accommodate the learning of feature extractor F. It means that in those previous epoches, the perplexity scores are not capable of providing reliable probablistic relations between target and each source. This hurts the pseudo-labeling scheme and further spoils the adversary at the next alternative step. Empirically, this negative effect mostly attributes to the unstable predictions to target instances. Hence we utilize the moving average to calculate the perpelxity score for each target instance.

$$s(\boldsymbol{x}_{N_T}^{(t)}; F, D_j) = \frac{1}{N_T}\sum_{i}^{N_T}(D_j(F(\boldsymbol{x}_i^{(t)})))^2, \tag{35}$$

where N_T denotes the number of times that the target samples have been visited to train our model (one target mini-batch as the measurement unit); $x_{N_T}^t$ denotes the current target instance being considered.

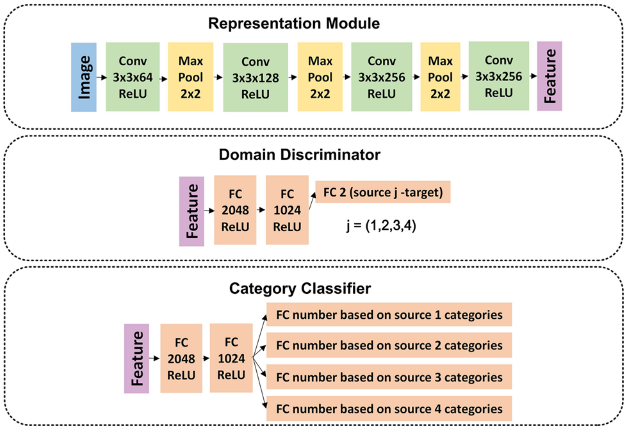

Fig. 12 The representation module, domain discriminator and category classifier we used in the experiments about digit recognition. (Best viewed in color)

Table 7 The hyper-parameters setting in our experiment

	Office-31	ImageCLE, DomainNet	Digit-five
Domain batch size	32	32	128
Threshold γ	0.9	0.98	0.9
Learning rate	0.00001	0.000002	0.00001
Image size	227×227	227×227	32×32
Virtual threshold ζ	top 15%	–,top 15%	–

Hyper-parameter setting of training In visual object recognition experiments (Office-31 and ImageCLEF), we initiate our DCTN by following the same way of DAN (Long et al. 2015). In terms of digit recognition, DCTN learns from scratch. In order to execute online hard domain mining, we construct our mini-batch by sampling an equal number of images per domain. For instance, consider a case of two-source domain adaptation with a domain batch size of 32. Then we have mini-batches with the sizes as $96 = 32 \times (2 + 1)$ (2 and 1 denote two source domains and one target domain). In this situation, the length of one epoch is decided by the size of the domain containing most instances. Finally, we adopt Adam (Kingma and Ba 2015) solver with $momentum = (0.9, 0.99)$ in all experiments to update our networks (Fig. 12).

More hyper-parameter details are shown in Table 7.

References

Baktashmotlagh, M., Harandi, M., & Salzmann, M. (2016). Distribution-matching embedding for visual domain adaptation. *The Journal of Machine Learning Research*, *17*(1), 3760–3789.

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

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Wortman, J. (2008). Learning bounds for domain adaptation. In *Advances in neural information processing systems* (pp. 129–136).

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 95–104).

Busto, P. P., & Gall, J. (2017). Open set domain adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 754–763).

Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018). Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2724–2732).

Cao, Z., Ma, L., Long, M., & Wang, J. (2018). Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision* (pp. 139–155).

Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2015). The cityscapes dataset. In *CVPR workshop on the future of datasets in vision*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 248–255).

Duan, L., Xu, D., & Tsang, I. W. H. (2012). Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(3), 504–518.

Fernando, B., Habrard, A., Sebban, M., & Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision* (pp. 2960–2967).

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning* (pp. 1180–1189).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2017). Domain-adversarial training of neural networks. In *Domain adaptation in computer vision applications* (p. 189).

Gebru, T., Hoffman, J., & Fei-Fei, L. (2017). Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *Proceedings of the IEEE international conference on computer vision* (pp. 1358–1367).

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., & Li, W. (2016). Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision* (pp. 597–613).

Gong, B., Grauman, K., & Sha, F. (2014). Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *International Journal of Computer Vision*, *109*(1–2), 3–27.

Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2066–2073).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gopalan, R., Li, R., & Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the IEEE international conference on computer vision* (pp. 999–1006).

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., & Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems* (pp. 513–520).

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., & Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, *3*(4), 5.

Haeusser, P., Frerix, T., Mordvintsev, A., & Cremers, D. (2017). Associative domain adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2784–2792).

Ho, H. T., & Gopalan, R. (2014). Model-driven domain adaptation on product manifolds for unconstrained face recognition. *International Journal of Computer Vision*, *109*(1–2), 110–125.

Hoffman, J., Wang, D., Yu, F., & Darrell, T. (2016). Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649

Jhuo, I. H., Liu, D., Lee, D., & Chang, S. F. (2013a). Robust visual domain adaptation with low-rank reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2168–2175).

Jhuo, I. H., Liu, D., Lee, D. T., & Chang, S. F. (2013b). Robust visual domain adaptation with low-rank reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2168–2175).

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. B. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1988–1997).

Kan, M., Wu, J., Shan, S., & Chen, X. (2014). Domain adaptation for face recognition: Targetize source domain bridged by common subspace. *International Journal of Computer Vision*, *109*(1–2), 94–109.

Kim, Y., Cho, D., & Hong, S. (2020). Towards privacy-preserving domain adaptation. *IEEE Signal Processing Letters*, *27*, 1675–1679.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*.

Koniusz, P., Tas, Y., & Porikli, F. (2017). Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7139–7148).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liang, X., Xu, C., Shen, X., Yang, J., Tang, J., Lin, L., et al. (2016). Human parsing with contextualized convolutional neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(1), 115–127.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *International conference on machine learning* (pp. 97–105).

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2016). Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems* (pp. 136–144).

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *Proceedings of the international conference on machine learning* (pp. 2208–2217).

Lu, H., Zhang, L., Cao, Z., Wei, W., Xian, K., Shen, C., & van den Hengel, A. (2017). When unsupervised domain adaptation meets tensor representations. In *Proceedings of the IEEE international conference on computer vision* (pp. 599–608).

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.

Mancini, M., Porzi, L., Bulò, S. R., Caputo, B., & Ricci, E. (2018). Boosting domain adaptation by discovering latent domains. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3771–3780).

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation with multiple sources. In *Advances in neural information processing systems* (pp. 1041–1048).

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2794–2802).

Motiian, S., Jones, Q., Iranmanesh, S. M., & Doretto, G. (2017). Few-shot adversarial domain adaptation. In *Advances in neural information processing systems* (pp. 6670–6680).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Nips workshop on deep learning and unsupervised feature learning*.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, *22*(2), 199–210.

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

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1406–1415).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *Proceedings of the European conference on computer vision* (pp. 213–226).

Saito, K., Ushiku, Y., & Harada, T. (2017). Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the international conference on machine learning* (pp. 2988–2997).

Saito, K., Yamamoto, S., Ushiku, Y., & Harada, T. (2018). Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision* (pp. 156–171).

Shao, M., Kit, D., & Fu, Y. (2014). Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, *109*(1–2), 74–93.

Sun, B., Feng, J., & Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *AAAI conference on artificial intelligence* (pp. 2058–2065).

Tzeng, E., Hoffman, J., Darrell, T., & Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4068–4076).

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2962–2971).

Xie, J., Hu, W., Zhu, S. C., & Wu, Y. N. (2015). Learning sparse frame models for natural image patterns. *International Journal of Computer Vision*, *114*(2–3), 91–112.

Xu, J., Ramos, S., Vázquez, D., & López, A. M. (2016). Hierarchical adaptive structural SVM for domain adaptation. *International Journal of Computer Vision*, *119*(2), 159–178.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Xu, R., Chen, Z., Zuo, W., Yan, J., & Lin, L. (2018). Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3964–3973).

Xu, R., Li, G., Yang, J., & Lin, L. (2019). Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1426–1435).

Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., & Zuo, W. (2017). Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 945–954).

Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain video concept detection using adaptive svms. In *Proceedings of the ACM international conference on multimedia* (pp. 188–197).

Yao, Y., Zhang, Y., Li, X., & Ye, Y. (2019). Heterogeneous domain adaptation via soft transfer network. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 1578–1586).

You, K., Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2019). Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2720–2729).

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., & Saminger-platz, S. (2017). Central moment discrepancy (cmd) for domain-invariant representation learning. In *International conference on learning representations*.

Zhang, J., Li, W., & Ogunbona, P. (2017). Joint geometrical and statistical alignment for visual domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5150–5158).

Zhang, S., Huang, J. B., Lim, J., Gong, Y., Wang, J., Ahuja, N., & Yang, M. H. (2019). Tracking persons-of-interest via unsupervised representation adaptation. *International Journal of Computer Vision,* 1–25.

Zhao, H., Zhang, S., Wu, G., Costeira, J. P., Moura, J. M. F., & Gordon, G. J. (2018). Multiple source domain adaptation with adversarial learning. In *International conference on learning representations*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.