Deep Residual Correction Network for Partial Domain Adaptation

Shuang Li, Chi Harold Liu, *Senior Member, IEEE,* Qiuxia Lin, Qi Wen, Limin Su, Gao Huang and Zhengming Ding

Abstract—Deep domain adaptation methods have achieved appealing performance by learning transferable representations from a well-labeled source domain to a different but related unlabeled target domain. Most existing works assume source and target data share the identical label space, which is often difficult to be satisfied in many real-world applications. With the emergence of big data, there is a more practical scenario called *partial domain adaptation*, where we are always accessible to a more large-scale source domain while working on a relative small-scale target domain. In this case, the conventional domain adaptation assumption should be relaxed, and the target label space tends to be a subset of the source label space. Intuitively, reinforcing the positive effects of the most relevant source subclasses and reducing the negative impacts of irrelevant source subclasses are of vital importance to address partial domain adaptation challenge. This paper proposes an efficiently-implemented *Deep Residual Correction Network* (DRCN) by plugging one residual block into the source network along with the task-specific feature layer, which effectively enhances the adaptation from source to target and explicitly weakens the influence from the irrelevant source classes. Specifically, the plugged residual block, which consists of several fully-connected layers, could deepen basic network and boost its feature representation capability correspondingly. Moreover, we design a weighted class-wise domain alignment loss to couple two domains by matching the feature distributions of shared classes between source and target. Comprehensive experiments on partial, traditional and fine-grained cross-domain visual recognition demonstrate that DRCN is superior to the competitive deep domain adaptation approaches.

Index Terms—Deep Transfer Leaning, Partial Domain adaptation, Maximum Mean Discrepancy, Fine-grained Visual Recognition.

1 Introduction

Numerous recent advances have dramatically improved the performance of deep neural networks (DNNs) at several diverse learning tasks such as network designing [1], [2], [3], [4], computer vision [5], [6], [7], [8], [9], and natural language processing [10], [11], [12], etc. Nevertheless, most state-of-the-art models with appealing performance mainly rely on the massive amount of annotated data, however, it is always time-consuming and expensive to obtain sufficient amount of labeled training data [13], [14], [15]. At the same time, traditional machine learning methods often have one common assumption that the training and test data are drawn from the same or similar probability distribution, which cannot always be satisfied in the real-world applications. Therefore, there is a strong motivation to leverage the useful knowledge of a related labeled *source domain* to help design versatile models for our interested unlabeled *target domain* following different feature distributions. To achieve this, domain adaptation [13], [16] is a promising strategy to address the domain shift issue, and impressive progress has been made in a wide range of scenarios [17], [18], [19], [20], [21], [22], [23], [24].

The mainstream methods of domain adaptation are either instance reweighting based methods [25], [26] by assigning different weights to labeled source samples, or domain-invariant feature learning based methods [27], [28], [29] by mitigating the distributions discrepancy across two domains. On the contrary, recent advance in deep learning reveals that deep architectures can extract more transferable and domain-invariant representations when dealing with domain adaptation problems [21], [30], [31]. Moreover, deep learning based methods have achieved superior performance over most shallow domain adaptation approaches.

However, most of the current deep domain adaptation methods still assume the source and target domains share one identical label space. In other words, data in both domains belong to exactly the same classes. Then, by aligning statistic moments [17], [21], [32], [33], [34] or leveraging adversarial techniques [18], [35], [36], [37], they could mitigate the marginal distribution differences across two domains to learn transferable representations. For feasible adaptation, the label spaces of both domains are required to be identical, otherwise the data in the irrelevant source subclasses will mislead the target data alignment and cause *negative transfer* when recognizing target classes.

Most recently, a more challenging and practical scenario, referred to as *partial domain adaptation*, attracts a lot of attentions [38], [39], [40], where a large-scale source domain is diverse enough to subsume all classes in a small-scale target domain of interest. Furthermore, target data are unlabeled and we have no idea about the size of target classes or the corresponding categories. Intuitively, to address partial

- *S. Li, C. H. Liu, Q. Lin, Q. Wen and L. Su are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Corresponding author: C. H. Liu. Email: {shuangli,chiliu,qlin,qwen,sulimin}@bit.edu.cn*
- *G. Huang is with Department of Automation, Tsinghua University, Beijing, China. Email: gaohuang@tsinghua.edu.cn*
- *Z. Ding is with Department of Computer, Information and Technology, Indiana University-Purdue University Indianapolis, USA. Email: zd2@iu.edu*

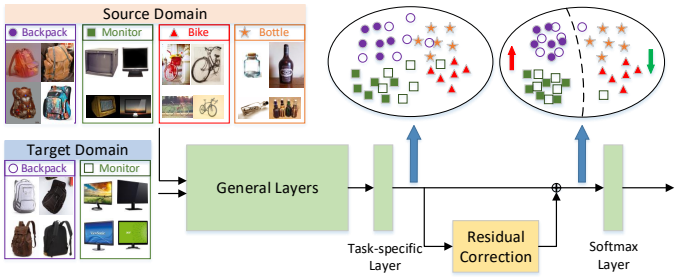


Fig. 1. Illustration of the proposed *Deep Residual Correction Network* (DRCN) approach that aims to address partial domain adaptation problem, where the label space of the interested small-scale target domain is a subset to that of a large-scale source domain. By utilizing the weighted class-wise matching, DRCN can effectively select out the irrelevant source classes and enhance the importance scores of the most relevant source classes. In addition, the plugged residual block could adaptively align source and target domains, and boost its feature representation capability, which results in better target classification performance.

domain adaptation problems, we cannot simply align the whole source and target domains, since the irrelevant source subclasses will be mixed with target data together, resulting in the degradation of target classification performance. Thus, moving out the irrelevant source classes and enhancing the effects of the most similar source classes with target domain are crucial for effective knowledge transfer. To achieve this purpose, [38], [39] propose to maximally match both domains' distributions in the shared label space and diminish the negative impact of irrelevant source classes. [40] designs a novel adversarial domain classifier to identify the importance weight of each source data automatically. However, these methods all utilize adversarial techniques to align source and target, which are difficult to optimize comparing with MMD loss based domain adaptation methods.

In this paper, we propose an efficiently-implemented *Deep Residual Correction Network* (DRCN) to address partial domain adaptation problem by identifying desirable classes and strengthening feature transfer. On the one hand, we present weighted class-wise domain alignment to automatically uncover the most relevant source classes to target data according to the target output probability distribution, and assign larger weights to maximally facilitate the class-wise alignment between these shared classes across domains. Besides, since the deep features in standard CNN architectures transit from general to specific along the network, the cross-domain discrepancy between source and target will increase, which results in the transferability decreasing of task-specific layers [31]. Therefore, we expect to enhance the domain shift mitigation between the two domains through the designed residual correction block. As illustrated in Fig. 1, we add one residual correction block along with the task-specific feature layer in a general network. Contributed by the class-wise alignment and enhancement of residual block, DRCN can maximally reduce the disparity of target data with those source data whose classes are likely to appear in the target label space.

On the other hand, when classifier prediction is quite poor in hard tasks, we believe domain-wise knowledge containing general information is also necessary in partial domain adaptation scenarios. To achieve this goal, DRCN aligns the joint distributions of last two domain-specific layers across domains by minimizing a joint maximum mean discrepancy (JMMD) metric [17], thereby greatly transferring the general feature-level and task-level knowledge from source to target. To sum up, we have four-fold contributions as follows:

- We propose an easily implemented but effective *Deep Residual Correction Network* (DRCN) to address the partial domain adaptation problem, which faces more realistic and challenging scenarios. DRCN only requires to plug one residual block into a unified network to mitigate the cross-domain distributions discrepancy. The added residual block can explicitly capture the feature difference between source and target, which intrinsically addresses the inherent problem in partial domain adaptation scenarios.
- DRCN introduces a weighted class-wise alignment loss which can automatically identify the most relevant source subclasses to target data based on the target output probability distribution. Larger weights will be assigned to these classes to benefit the accurate class-wise matching across domains a lot.
- Moreover, DRCN properly explores domain-wise knowledge to adapt global source information to target, which is a different attempt from the existing partial domain adaptation methods. When a large cross-domain discrepancy exists, the leveraging of domain-wise knowledge is crucial for effectively solving partial domain adaptation problems.
- Furthermore, DRCN can be easily extended to deal with traditional and fine-grained cross-domain visual recognition, in which source and target domains share one identical label space or even fined-grained visual data. This manifests the universality of the proposed DRCN. Extensive experiments on partial, traditional and fine-grained cross-domain visual recognition demonstrate DRCN outperforms other competitive comparisons with a large margin.

2 Related Work

Recent progresses in domain adaptation [13] are able to address the issue that training and test data follow different feature spaces and data distributions. This helps mitigate the burden of manual labeling by exploring the external source knowledge, which promotes impressive research efforts in lots of applications [16], [18], [38], [39], [41], [42].

2.1 Domain Adaptation

Extensive prior works on domain adaptation usually attempt to minimize the domain discrepancy through instance reweighting or domain-invariant feature learning [13]. Instance reweighting based methods are encouraged to align source and target distribution by reweighting source samples [19], [25]. However, these methods exist limitation when training and test data follow different conditional distributions. On the other hand, feature learning based methods aim to derive domain-invariant features or latent subspaces, by reducing the distribution differences across domains. For example, Scatter Component Analysis (SCA, [43]) seeks to trade between maximizing the separability of

classes, minimizing the mismatch between domains, and maximizing the separability of data. TCA [16] considers learning some transfer components in RKHS to minimize the distance of source and target marginal distributions using Maximum Mean Discrepancy (MMD) metric. Benefiting from leveraging target pseudo labels, JDA [27] attempts to match both marginal and conditional distributions of two domains by requiring their total means and class means to be close to each other. However, these methods all can only learn shallow features for both domains, which are not effective enough comparing with deep learning based domain adaptation methods.

2.2 Deep Domain Adaptation

Deep learning can learn abstract representations that disentangle different explanatory factors of variations behind data [44] and manifest invariant factors underlying different populations that transfer well across similar tasks [31]. However, some recent findings reveal that deep networks can only reduce, but not eliminate, the cross-domain discrepancy [31], [32].

Hence, there are several recent attempts in bridging deep learning with domain adaptation, which are generally achieved by adding adaptation layers through which the means of distributions are matched [32], [34] or adding a subnetwork as a domain discriminator which would be confused by learned deep features from two domains [18], [36]. Some previous works on deep domain adaptation usually attempt to align two domains by utilizing MMD metric. To name a few, Tzeng *et al.* in [32] utilize an adaptation layer along with a domain confusion loss based on MMD to automatically learn the domain-invariant representations jointly trained with a classifier. In [21], Long *et al.* leverage multi-layer adaptation via multi-kernel MMD, which is able to align different domains in the task-specific layers. Being primarily motivated by ResNet in [45], [34] proposes Residual Transfer Networks (RTN) to bridge the source classifier $f_s(x)$ and target classifier $f_t(x)$ with the residual layers, such that the classifier mismatch across domains can be explicitly modeled by the residual functions $\Delta F(x)$ in a deep learning architecture. Different from RTN, our proposed residual block aims to explicitly learn and mitigate the feature difference, instead of the classifier difference, between source and target, which can intrinsically address the inherent problem in domain adaptation scenarios. In addition, we find that it is more effective to mitigate the domain feature discrepancy rather than classifier difference. Furthermore, Joint Adaptation Networks (JAN) [17] develops a joint maximum mean discrepancy (JMMD) criterion which can be calculated by back-propagation in linear-time.

Another line of methods are based on an adversarial loss which introduces a novel domain classifier to promote domain confusion, where the data would be indiscriminative with respect to domain labels. Recent works [18], [36], [37], combining adversarial learning with domain adaptation, have shown significantly improved performance. Specifically, [37] outlines a novel generalized adversarial adaptation framework, which subsumes several deep adversarial transfer learning methods as special cases. Based on this framework, [37] also develops an Adversarial Discriminative Domain Adaptation (ADDA) approach. By exploring the adversarial loss in aligning two domains in terms of feature level or image level, there are many research efforts done recently [18], [36], [46], [47]. The general idea behind is to generate domain-invariant features across two domains by confusing the discriminator. Usually minimax strategy is adopted to optimize two players in adversarial learning based methods, thus it is usually hard to achieve stable solutions compared with MMD-based methods.

However, these methods all assume the source and target label spaces are the same, which is often too strict to be satisfied in real-world applications. The proposed DRCN has relaxed this common assumption, and aims to address a more challenging partial domain adaptation problem.

2.3 Partial Domain Adaptation

In reality, we are much easier to obtain a large-scale source dataset, while working on a small-scale unlabeled target dataset, which requires us to transfer partial relevant knowledge from the source to target. In this way, previous domain adaptation methods [21], [37], typically assuming that source and target domains have identical label space, are prone to *negative transfer* for the partial transfer problems.

To effectively address partial domain adaptation problems, Cao *et al.* in [39] presents a Selective Adversarial Network (SAN), which can promote positive transfer of relevant data and alleviate negative transfer of irrelevant data simultaneously, and maximally match the data distributions in the shared class space. Partial Adversarial Domain Adaptation (PADA) [38] alleviates the negative transfer by down-weighing the data of outlier source classes. Zhang *et al.* propose a novel adversarial nets-based partial domain adaptation method to identify the source samples that are potentially from the outlier classes, while reducing the domain shift of shared classes between domains [40]. These methods all rely on various adversarial strategies and networks to learn the target model.

Different from them, our DRCN only plugs one residual correction block into source network to explicitly mitigate the domain discrepancy between domains. By leveraging the target output probability distribution, DRCN could effectively identify the most relevant source subclasses and maximally conduct feature alignment through the proposed weighted class-wise matching scheme, which is clearly different from other partial domain adaptation methods.

3 The Proposed Algorithm: DRCN

This section introduces our proposed Deep Residual Correction Network (DRCN) in detail. The background and preliminary knowledge will be presented first.

3.1 Preliminary

Partial domain adaptation is a novel but practical transfer learning paradigm [38], [39], which manages to transfer relevant knowledge from a large-scale source domain to a small-scale target domain. Similar to traditional domain adaptation terminologies, in partial domain adaptation, labeled source domain $\mathcal{D}_s$ and unlabeled target domain $\mathcal{D}_t$ are provided, where $\mathcal{D}_s = \{(\boldsymbol{x}_{si}, y_{si})\}_{i=1}^{n_s} = \{\mathbf{X}_s, \boldsymbol{y}_s\}$,

$\mathcal{D}_t = \{\boldsymbol{x}_{tj}\}_{j=1}^{n_t} = \{\mathbf{X}_t\}$. y_{si} is the corresponding label of $\boldsymbol{x}_{si}$. n_s, n_t are the numbers of source and target samples.

Suppose $\mathcal{X}_s, \mathcal{X}_t$ and $\mathcal{Y}_s, \mathcal{Y}_t$ are the feature and label spaces of source and target domains, respectively. For partial domain adaptation, $\mathcal{X}_s = \mathcal{X}_t$, while the target label space $\mathcal{Y}_t$ is a subset of the source label space $\mathcal{Y}_s$, i.e., $\mathcal{Y}_t \subseteq \mathcal{Y}_s$. Here we denote C_s, C_t as the class numbers of source and target domain, and $C_s > C_t$. By contrast, C_s equals to C_t in the traditional domain adaptation scenario. In addition, due to the domain shift, the source and target feature distributions $P_s(\boldsymbol{x}) \neq P_t(\boldsymbol{x})$. The goal is to maximally mitigate the partial distribution discrepancy across domains, and transfer the most relevant source discriminative knowledge to target domain effectively.

3.2 Motivation

Since partial transfer learning assumes target label space subsumes to source label space, and the distributions of both domains with the shared classes are also different. It is of vital importance to figure out *what knowledge should be transferred from a large-scale source to a small-scale target*, and *how to capture the relevant knowledge effectively*?

Intuitively, when the categories of target data are only a subset of source data, we should avoid only matching the distributions of the whole source and target domains, since the irrelevant source classes will have negative effect on the domain alignment. As a result, the most relevant source classes to target domain should be effectively identified, and precisely align their conditional distributions between two domains. Nevertheless, we also believe the generic information learned by pre-trained network on massive source data is valuable and crucial for partial domain adaptation. Additionally, because task-specific layer contains feature-level knowledge while classifier layer contains rich multi-modal structure which is task-level knowledge, we therefore want to simultaneously align these layers to greatly transfer general feature-level and task-level knowledge. To achieve the aforementioned goals, we propose to align the joint distributions between input features and output labels across domains, as well as the class-wise distributions between source and target in the shared class space.

Recent works of deep domain adaptation have revealed that many consecutive layers of non-linear transformations within the trained source network will amplify the feature distributions difference across domains [21], [31]. Thus, an effective and direct way to compensate the domain shift is to correct the feature discrepancy of source and target right along the task-specific layer. Inspired by the well-known residual network [45], we aim to plug one residual correction block, consisting of several layers, into the trained source network to explicitly learn the feature difference between source and target domains. In addition, the added residual block could improve the generalization ability of the infrastructural network by deepening it. This small modification of source network has been proven to be very effective in our experiments.

3.3 Deep Residual Correction Network

We first revisit maximum mean discrepancy (MMD) [48] metric to facilitate the feature adaptation across domains, and then present the details of our proposed DRCN.

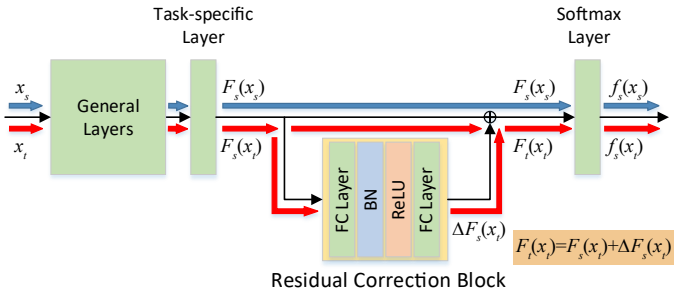


Fig. 2. The architecture and data flow of the added residual correction block in a pre-trained source network, where a FC Layer is a fully-connected layer with appropriate dimensions, BN represents the Batch Normalization layer [52], and ReLU is the non-linear transformation. The blue and red arrows denote the data flow of source and target data, respectively. DRCN expects the added residual correction block to learn the difference $\Delta F_s(\boldsymbol{x}_t)$ between feature representations $F_s(\boldsymbol{x}_s)$ and $F_s(\boldsymbol{x}_t)$, such that $F_s(\boldsymbol{x}_s)$ and $F_t(\boldsymbol{x}_t)$ can be more similar.

3.3.1 Maximum Mean Discrepancy Revisit

MMD statistical test is commonly leveraged to quantitatively measure the similarity of source and target probability distributions $P_s(\boldsymbol{x})$ and $P_t(\boldsymbol{x})$ [17], [21], [34]. If we denote $\mathcal{F}$ is a universal class of functions, MMD can be formally represented as:

$$\mathcal{D}_{\mathrm{MMD}}(P_s, P_t) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{\boldsymbol{x} \sim P_s}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x} \sim P_t}[f(\boldsymbol{x})]\right)^2, \quad (1)$$

where f is a function from $\mathcal{F}$. It has been theoretically proven that $P_s(\boldsymbol{x})$ and $P_t(\boldsymbol{x})$ are identical if and only if $\mathcal{D}_{\mathrm{MMD}}(P_s, P_t) = 0$ [48].

If $\mathcal{F}$ is a universal RKHS with kernel function $\kappa(\cdot, \cdot)$, (1) can be represented as

$$\begin{aligned}\mathcal{D}_{\mathrm{MMD}}(P_s, P_t) &= \mathbb{E}_{\boldsymbol{x}_s, \boldsymbol{x}_s' \sim P_s}[\kappa(\boldsymbol{x}_s, \boldsymbol{x}_s')] \\ &- 2\mathbb{E}_{\boldsymbol{x}_s \sim P_s, \boldsymbol{x}_t \sim P_t}[\kappa(\boldsymbol{x}_s, \boldsymbol{x}_t)] + \mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{x}_t' \sim P_t}[\kappa(\boldsymbol{x}_t, \boldsymbol{x}_t')]. \quad (2)\end{aligned}$$

In (2), $\boldsymbol{x}_s, \boldsymbol{x}_s'$ and $\boldsymbol{x}_t, \boldsymbol{x}_t'$ are samples drawn from distributions $P_s(\boldsymbol{x})$ and $P_t(\boldsymbol{x})$ respectively. When we have finite source and target samples, the MMD distance of both domains can be estimated as

$$\begin{aligned}\widehat{\mathcal{D}}_{\mathrm{MMD}}(P_s, P_t) &= \frac{1}{n_s^2} \sum_{i,j=1}^{n_s} \kappa(\boldsymbol{x}_{si}, \boldsymbol{x}_{sj}) \\ &- \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \kappa(\boldsymbol{x}_{si}, \boldsymbol{x}_{tj}) + \frac{1}{n_t^2} \sum_{i,j=1}^{n_t} \kappa(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj}) \quad (3)\end{aligned}$$

MMD has been successfully applied in massive areas of deep learning, ranging from generative adversarial models [49], [50] to image transformation [51]. The general idea behind using MMD in this paper is to explicitly depict the distribution difference between source and target domains. By minimizing the improved MMD metric [17], [21], we could achieve effective feature and task adaptation.

3.3.2 Residual Correction Block

Deep architectures can learn more abstract, transferable and desirable features automatically [21], [31]. However, the learned deep features can only reduce, but not remove, the substantial cross-domain discrepancy. Especially, the consecutive non-linear transformation in a deep network may amplify the difference between domains.

One intuitive way to mitigate the cross-domain discrepancy is to correct it right after the task-specific layer of a source network, since recent studies have revealed that the feature transferability drops from general feature extraction layers to higher task-specific layers under the domain discrepancy [31]. Meanwhile, the features learned in the higher task-specific layers are more crucial for the final classification. Therefore, as shown in Fig. 2, after the task-specific layer in a pre-trained source network, we plug one residual correction block to explicitly learn the cross-domain discrepancy. Most works align features in task-specific layer just by deploying discrepancy losses, whereas we find that exploiting the structure modification can further explicitly measure and learn the difference. Our designed residual correction block will not only keep the feature extraction and discriminative classification abilities of the original source network, but also benefit the deep adaptation from source to target of interest.

As illustrated in Fig. 2, we denote the task-specific features of source data $\boldsymbol{x}_s$ and target data $\boldsymbol{x}_t$ as $F_s(\boldsymbol{x}_s)$ and $F_s(\boldsymbol{x}_t)$, respectively. $\boldsymbol{x}_s$ only goes through the original source network. Besides the source network, $\boldsymbol{x}_t$ also passes the added residual correction block, which aims to learn the domain discrepancy $\Delta F_s(\boldsymbol{x}_t)$. Then, we expect that the modified target representations $F_t(\boldsymbol{x}_t) = F_s(\boldsymbol{x}_t) + \Delta F_s(\boldsymbol{x}_t)$ will be much more similar to source representations $F_s(\boldsymbol{x}_s)$ after the adaptation. Because we only let target sample pass through residual correction block and align it with source sample, such a weakly-shared structure allows target domain to model its difference with source domain effectively and properly. Besides, the added residual block deepens the general network, which could improve its feature representation capability correspondingly.

To this end, in DRCN, we first require the empirical error of the source network classifier f_s on labeled source data $\{(\boldsymbol{x}_{si}, y_{si})\}_{i=1}^{n_s}$ to be minimized:

$$\min_{f_s} \quad \mathcal{L}_s = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathcal{L}(f_s(\boldsymbol{x}_{si}), \boldsymbol{y}_{si}), \tag{4}$$

where $\boldsymbol{y}_{si}$ is the one-hot label for $\boldsymbol{x}_{si}$ and $\mathcal{L}(\cdot,\cdot)$ is the cross-entropy loss function defined as follows:

$$\mathcal{L}(f_s(\boldsymbol{x}_{si}), \boldsymbol{y}_{si}) = -\sum_{k=1}^{C_s}\mathbf{1}_{[k=y_{si}]}\log {f_s}^{(k)}(\boldsymbol{x}_{si}), \tag{5}$$

where ${f_s}^{(k)}(\boldsymbol{x}_{si})$ is the probability of $\boldsymbol{x}_{si}$ predicted to the k-th class by f_s.

Similar to [17], [21], [40], we adapt the pre-trained source network from source data to a small-scale target domain. For partial domain adaptation, we also need to select out the irrelevant source subclasses, and assign larger importance scores to these most relevant source data. To achieve this goal, DRCN could automatically calculate the importance of each source class according to the target output probability distribution, and explicitly conduct weighted class-wise distribution matching to transfer the most related and useful knowledge from source to target.

3.3.3 Weighted Class-wise Alignment

Different from the traditional domain adaptation settings, where the label spaces of both domains are identical, partial domain adaptation assumes target domain only contains a subset of classes in source domain. Therefore, identifying the most relevant source subclasses to target, and conducting class-wise feature distributions alignment of shared classes across domains are crucial to derive perfect partial domain adaptation models.

In this paper, we leverage the target output probability distribution, referred to as target "soft label" [36] in each category, predicted by the source classifier, to identify the most possible classes that target data contain. Then, we can select the corresponding source subclasses to conduct class-wise matching. To be specific, for each target data $\boldsymbol{x}_{tj}$, its prediction by the source classifier f_s is denoted as $\boldsymbol{p}_{tj}$. $\boldsymbol{p}_{tj}$ is a label vector, whose k-th element $\boldsymbol{p}_{tj}^{(k)}$ represents the probability of assigning $\boldsymbol{x}_{tj}$ to class k in C_s classes. If we average all target predictions from f_s, we can obtain the target output probability distribution over C_s classes, which could effectively uncover the proportion of each target class in a larger source label space.

The average of all the target predictions could be computed as:

$$\boldsymbol{w} = \frac{1}{n_t}\sum_{j=1}^{n_t}\boldsymbol{p}_{tj}, \tag{6}$$

which could be used to assign different importance scores to different classes. From (6), we observe that if the kth element of $\boldsymbol{w}$, i.e. $\boldsymbol{w}^{(k)}$ is sufficiently small, it means all the target data are predicted to class k with small probability, and the class k of source data should be irrelevant to the small-scale target domain with large probability. Thus, we should assign a small weight to class k when we conduct class-wise feature alignment, and vice versa. In practice, we normalize the weight $\boldsymbol{w}$ by dividing its maximum element as [38], i.e.

$$\boldsymbol{w} \leftarrow \frac{1}{\max(\boldsymbol{w})}\boldsymbol{w}. \tag{7}$$

After identifying the most relevant source subclasses automatically, we conduct weighted class-wise distribution matching to improve the positive effects of relevant source classes and reduce the negative impacts of irrelevant source classes, which could benefit transferring valuable knowledge from source to target a lot.

To be specific, we leverage multiple kernel variant of MMD (MK-MMD) [21], [48] to effectively reduce the discrepancy of each class across domains. The squared formulation of MK-MMD distance for source and target data representations of class k can be estimated as:

$$\widehat{\mathcal{D}}_{\mathrm{MK-MMD}}^{(k)} = \left\| \frac{1}{n_s^{(k)}}\sum_{\boldsymbol{x}_{si}\in\mathcal{D}_s^{(k)}}\phi\big(F_s(\boldsymbol{x}_{si})\big) - \frac{1}{n_t^{(k)}}\sum_{j=1}^{n_t}\boldsymbol{p}_{tj}^{(k)}\phi\big(F_t(\boldsymbol{x}_{tj})\big)\right\|_{\mathcal{H}_\kappa}^2, \tag{8}$$

where $n_t^{(k)} = \sum_{j=1}^{n_t}\boldsymbol{p}_{tj}^{(k)}$, $\mathcal{D}_s^{(k)}$ denotes all the source samples with their true labels being k, $\mathcal{H}_\kappa$ is the RKHS with a characteristic kernel κ. ϕ is the corresponding feature map.

For partial domain adaptation problems, we should apply the class weight, computed in (7) to improve the contributions of relevant classes adaptation between two domains, and down-weigh the contributions of irrelevant

classes alignment when learning the final model. Therefore, the loss term of weighted class-wise distribution matching in DRCN can be formulated as

$$\mathcal{L}_{class} = \sum_{k=1}^{C_s} \boldsymbol{w}^{(k)} \cdot \widehat{\mathcal{D}}^{(k)}_{\mathrm{MK-MMD}}. \tag{9}$$

By minimizing (9), DRCN could put more emphasis on the shared classes with larger weights, and transfer the valid feature and task knowledge from source to target without distraction of irrelevant source classes. Different from other partial domain adaptation [38], [39], [40], we propose to explicitly match class-wise distributions across domains in the shared label space by minimizing a novel probabilistic class-wise multiple kernel MMD, which considers prior target category distributions to reduce the conditional distribution divergence in an effective way.

However, we find that general information is also crucial for partial domain adaptation, as class-wise alignment might not work well when model predication is very poor. Therefore, to maximally preserve the characteristics and general knowledge of a large-scale source domain, we opt to align features and classifiers of both domains in a general way, which facilitates transferring the basic knowledge from source to target.

3.3.4 General Feature Adaptation

It is noteworthy that, in contrast to other partial domain adaptation methods [38], [39], [40], DRCN also aligns the joint distributions of both domains to some extent. [38], [39], [40] claim that the irrelevant source subclasses will degrade the adaptation performance, and thus, they do not intend to align the whole source and target domains.

Nevertheless, we believe the general feature-level and task-level source knowledge are also of great importance to unlabeled target domain. Since the classifier may produce wrong predictions when tasks are difficult, the class-wise alignment cannot greatly favor learning discriminative features, and accordingly we need general knowledge transfer. To maximally learn general knowledge, we want to align task-specific layer containing feature-level knowledge and classifier layer containing rich multimodal structure which is task-level knowledge. However, we cannot align feature and class distribution separately, since the multimodal structures can only be captured sufficiently by the cross-covariance dependency between features and classes [53]. To address this problem, [17] proposes joint maximum mean discrepancy (JMMD) which can achieve cross-covariance dependency of feature representation and class prediction by multiplying interactions between them. We therefore minimize the JMMD distance of multiple feature layers (task-specific layer and softmax layer) across domains to enable safe and effective knowledge transfer. The key point is to control the importance of this part.

To be specific, we denote the activations of n_s source data generated by the task-specific layer and softmax layer as $\{F_s(\boldsymbol{x}_{si})\}_{i=1}^{n_s}$ and $\{f_s(\boldsymbol{x}_{si})\}_{i=1}^{n_s}$, respectively. All the target data will go through the source task-specific layer and residual correction block simultaneously. The activations of n_t target data generated by the element-wise summation and the softmax layer of source network as $\{F_t(\boldsymbol{x}_{tj})\}_{j=1}^{n_t}$ and $\{f_s(\boldsymbol{x}_{tj})\}_{j=1}^{n_t}$. To adapt the two layers effectively, we leverage the tensor product between feature-level and task-level activations to conduct joint distributions alignment.

Fig. 3. Images examples of datasets Office-Home, Office-31, VisDA2017 and ImageNet-Caltech.

Generally, the empirical estimate of JMMD between source and target domains in DRCN can be represented as:

$$\begin{aligned}\mathcal{L}_{domain} &= \widehat{\mathcal{D}}_{\mathrm{JMMD}}(P_s, P_t) \\ &= \frac{1}{n_s^2}\sum_{i,j=1}^{n_s} \kappa_1(F_s(\boldsymbol{x}_{si}), F_s(\boldsymbol{x}_{sj})) \cdot \kappa_2(f_s(\boldsymbol{x}_{si}), f_s(\boldsymbol{x}_{sj})) \\ &\quad - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} \kappa_1(F_s(\boldsymbol{x}_{si}), F_t(\boldsymbol{x}_{tj})) \cdot \kappa_2(f_s(\boldsymbol{x}_{si}), f_s(\boldsymbol{x}_{tj})) \\ &\quad + \frac{1}{n_t^2}\sum_{i,j=1}^{n_t} \kappa_1(F_t(\boldsymbol{x}_{ti}), F_t(\boldsymbol{x}_{tj})) \cdot \kappa_2(f_s(\boldsymbol{x}_{ti}), f_s(\boldsymbol{x}_{tj})),\end{aligned} \tag{10}$$

where $\kappa_1(\cdot,\cdot)$ and $\kappa_2(\cdot,\cdot)$ are the corresponding kernel functions. In DRCN, we choose Gaussian kernel function as the kernel function. In practical, to accelerate the computation speed of (10), we adopt a linear-time estimate of JMMD as follows:

$$\begin{aligned}&\mathcal{L}_{domain} = \\ &\frac{2}{n}\sum_{i=1}^{n/2}\Big(\kappa_1(F_s(\boldsymbol{x}_{s2i-1}), F_s(\boldsymbol{x}_{s2i})) \cdot \kappa_2(f_s(\boldsymbol{x}_{s2i-1}), f_s(\boldsymbol{x}_{s2i})) \\ &\qquad + \kappa_1(F_t(\boldsymbol{x}_{t2i-1}), F_t(\boldsymbol{x}_{t2i})) \cdot \kappa_2(f_s(\boldsymbol{x}_{t2i-1}), f_s(\boldsymbol{x}_{t2i})) \\ &\qquad - \kappa_1(F_s(\boldsymbol{x}_{s2i-1}), F_t(\boldsymbol{x}_{t2i})) \cdot \kappa_2(f_s(\boldsymbol{x}_{s2i-1}), f_s(\boldsymbol{x}_{t2i})) \\ &\qquad - \kappa_1(F_t(\boldsymbol{x}_{t2i-1}), F_s(\boldsymbol{x}_{s2i})) \cdot \kappa_2(f_s(\boldsymbol{x}_{t2i-1}), f_s(\boldsymbol{x}_{s2i}))\Big).\end{aligned} \tag{11}$$

To calculate (11) during training, a minibatch of samples can be divided into quad-tuples, each containing two source samples and two target samples. This linear estimation of JMMD enables us to deal with large datasets effectively.

3.3.5 Overall Formulation of DRCN

To enable effective partial domain adaptation, we propose a Deep Residual Correction Network (DRCN), which aligns

TABLE 1
Statistics of the benchmark datasets.

Dataset	Sub-domain	Abbr.	Sample	#Class
Office-Home	Art	Ar	2427	65
	Clipart	Cl	4365	
	Product	Pr	4439	
	Real-World	Rw	4357	
Office-31	Amazon	A	2817	31
	DSLR	D	498	
	Webcam	W	795	
VisDA2017	Synthetic	S	152397	12
	Real	R	55388	
ImageNet-Caltech	Caltech-256	C	39037	256
	ImageNet-1K	I	1285367	1000

the feature distributions across domains in the shared label space with larger importance scores, while jointly adapting general feature-level and task-level knowledge from source to target. By incorporating Eq. (4), (11) and (9), the overall formulation of DRCN can be represented as:

$$\mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_{domain} + \beta \mathcal{L}_{class}, \tag{12}$$

where α and β are trade-offs to balance different terms.

Remark: First, the soft labels of target data in DRCN could well characterize the output distribution of target data, which help identify the most relevant source subclasses effectively. We refer to DRCN with soft labels as **DRCN (soft label)**. Here, the soft labels in Eq. (6) and Eq. (8) can also be replaced by hard labels, i.e., one-hot labels with the predicted class being 1. We define the hard label version as **DRCN (hard label)**. The pros and cons of the two schemes will be discussed in Section 4.

Second, it is worthy to note that DRCN can not only deal with partial domain adaptation (target classes are a subset of source classes), but also address traditional domain adaptation scenarios (source and target label spaces are identical), by setting $\boldsymbol{w}$ to be an all one vector, which means each class weight is 1 with equal importance contribution for class-wise alignment loss. Moreover, DRCN can be naturally extended to solve fine-grained cross-domain visual recognition, where source and target are both fine-grained images, e.g., different models of car.

4 Experiment

In this section, we compare our proposed DRCN with several competitive unsupervised domain adaptation to verify the superiority of DRCN. Besides, we further conduct several empirical experiments to demonstrate the flexibility and effectiveness of our approach. Our source code is released as https://github.com/wenqiwenqi1/DRCN.

We conduct the performance evaluation by using four cross-domain object recognition datasets: **Office-Home** [54], **Office-31** [55], **VisDA2017** and **ImageNet-Caltech**. Detailed information about these datasets is summarized in Table 1 and sample images per dataset are visualized in Figure 3.

Office-Home is released at CVPR'17, containing 65 different objects from 4 domains, as: Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**) and Real-World images (**Rw**). For each domain, the dataset consists of images found typically in Office and Home settings. Specifically, the images of domain **Ar** are paintings, sketches or artistic depictions, and that of **Cl** are clipart images. **Rw** comprises regular images captured by cameras, while the images in **Pr** are all without background. These images in the dataset are crawled through several search engines and online image directories, and widely used in domain adaptation algorithms [14], [38]. For these four domains, similar to [38], the first 25 classes (in alphabetical order) in that domain are taken as target categories, if one of them is selected as target domain. Considering all the domain combinations, we build 12 cross-domain learning tasks: **Ar**→**Cl**, ..., **Rw**→**Pr**.

Office-31 is widely adopted by deep transfer learning methods [14], [39], [56], with 4110 images that are commonly encountered objects in office settings, such as monitors, bottles and backpacks. It includes 31 classes and involves three distinct domains, as Amazon (**A**, which is downloaded from online merchants), Webcam (**W**) and DSLR (**D**). **W**, **D** contain images taken by web cameras and digital SLR cameras, respectively. We follow similar settings as [38], [39] and select 10 classes shared by Office-31 and Caltech-256 [57] as target categories. The corresponding images in Office-31 are defined as target domain. Hence, we can construct 6 domain adaptation transfer tasks: **A**→**W**, ..., **D**→**W**.

VisDA2017 is a large-scale dataset for cross-domain object classification which was first present in 2017 Visual Domain Adaptation (VisDA) Challenge. In the experiment, we use the training and validation images provided by the competition as 2 domains: one comprises synthetic 2D renderings of 3D models generated from different angles, and the other consists photo-realistic images or real images. Both of them have 12 categories in common. We denote the domain with synthetic images as **S**, while the domain with real images as **R**. Besides, we assume that if one domain is selected as target domain, the first 6 categories (in alphabetical order) will be chosen as target categories with corresponding images as target domain. Hence, we can conclude 2 transfer tasks: **S**→**R** and **R**→**S**.

ImageNet-Caltech is a challenging dataset which consists of ImageNet-1K and Caltech-256 and we denote them as **I** and **C** respectively. Comparing with former datasets, it is larger with over 1M images across 1000 classes in ImageNet-1K (**I**) and over 39K images across 256 classes in Caltech-256 (**C**). Since there are 84 common classes in both, we therefore build two partial transfer tasks: **I (1000)** → **C (84)** and **C (256)** → **I (84)**. And similar to [38], we use ImageNet-1K validation set as target domain when it comes to task **C (256)** → **I (84)**.

4.1 Baseline Methods

For the diversity of the experiment, we first take two shallow methods: GFK [58], TCA [16] with the deep features of ResNet-50 layer *pool5* as baselines. To extensively verify the effectiveness of DRCN, we compare DRCN with 12 traditional or partial deep domain adaptation methods: ResNet [45], DAN [21], RevGrad [18], RTN [34], JAN [17], LEL [59], ADDA [37], MADA [35], CDAN [60], SAN [39], IWAN [40], and PADA [38]. Note that several results are directly from the published papers if we follow the same setting.

4.2 Setup

We follow standard protocols as [21], in which the source data are all labeled while target data are unlabeled. Besides,

TABLE 2
Accuracy (%) on Office-Home for **partial** transfer learning tasks (ResNet-50).

Method	Office-Home												
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
ResNet [45]	38.6	60.8	75.2	39.9	48.1	52.9	49.7	30.9	70.8	65.4	41.8	70.4	53.7
DAN [21]	44.4	61.8	74.5	41.8	45.2	54.1	46.9	38.1	68.4	64.4	45.4	68.9	54.5
RevGrad [18]	44.9	54.1	69.0	36.3	34.3	45.2	44.1	38.0	68.7	53.0	34.7	46.5	47.4
RTN [34]	49.4	64.3	76.2	47.6	51.7	57.7	50.4	41.5	75.5	70.2	51.8	74.8	59.3
IWAN [40]	53.9	54.5	78.1	61.3	48.0	63.3	54.2	**52.0**	**81.3**	76.5	56.8	**82.9**	63.6
SAN [39]	44.4	68.7	74.6	**67.5**	65.0	**77.8**	59.8	44.7	80.1	72.2	50.2	78.7	65.3
PADA [38]	52.0	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.1
DRCN (soft label)	**54.0**	**76.4**	**83.0**	62.1	64.5	71.0	**70.8**	49.8	80.5	**77.5**	**59.1**	79.9	**69.0**
DRCN (hard label)	51.6	75.8	82.0	62.9	**65.1**	72.9	67.4	50.0	81.0	76.4	57.7	79.3	68.5

we average the accuracy of each cross-domain task by performing three random experiments and resize the input images to 256×256.

For all the experiments, DRCN and other compared deep domain adaptation methods are implemented with PyTorch, and fine-tuned from PyTorch-provided model of ResNet-50 pre-trained on ImageNet similar to the previous works [17], [38]. In this paper, we use the same learning rate as JAN [17] in classifier layer, which is 10 times larger than other layers. Since the residual layer is trained from scratch and it needs to be very precise, we set its learning rate to be one tenth that of the other layers. In addition, we use stochastic gradient descent (SGD) with momentum of 0.9 and a learning rate annealing strategy as [38]. We compute the class weights after each epoch when all target samples are trained. Therefore, the weights would be updated dynamically. For the fair comparison, we use the importance weighted cross-validation technique as [60] to select hyper-parameters, and set $\alpha = 0.1$ and $\beta = 0.05$ throughout all the experiments. Moreover, we will give parameter sensitivity analysis for DRCN, which indicates that for reasonable parameter values, DRCN always can achieve stable performance.

4.3 Results and Analysis

We evaluate DRCN on the following four datasets, Office-Home, Office-31, VisDA2017 and ImageNet-Caltech, by comparing with competitive traditional and partial deep domain adaptation methods, and make some discussions.

4.3.1 Partial Transfer Results on Office-Home

Table 2 summarizes the experimental results of Office-Home, in which we can observe that DRCN outperforms all comparison methods on most tasks. Note that some traditional domain adaptation methods, like DAN, RevGrad, achieve worse performance on most tasks compared with ResNet, which implies that only conducting distribution alignment across domains is not enough for partial domain adaptation problems. Whereas PADA beats ResNet by an average accuracy improvement of 8.4% while IWAN and SAN increase 1.5% and 3.2% respectively compared to PADA. It is reasonable since IWAN and SAN consider to alleviate negative transfer by detecting outlier classes with various weighting schemes. It is worthy to note that DRCN gains the highest accuracies on 7 out of 12 tasks, and promotes significantly on several domain transfer tasks, i.e., **Ar→Pr**, **Pr→Ar** and **Ar→Rw**. For average accuracy, DRCN works best by outperforming baseline SAN with at least **3.2%** improvement, indicating that DRCN is more conductive to weaken the influence from the irrelevant source classes thereby encouraging relevant information transfer among shared classes. Besides, soft label based DRCN is slightly better than hard label based DRCN, because soft label contains class structure information that can help correct prediction deviations.

TABLE 3
Accuracy (%) on Office-31 for **partial** transfer learning tasks (ResNet-50).

Method	Office-31						
	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet [45]	54.5	94.6	94.3	65.6	73.2	71.7	75.6
DAN [21]	46.4	53.6	58.6	42.7	65.7	65.3	55.4
RevGrad [18]	41.4	46.8	38.9	41.4	41.3	44.7	42.4
ADDA [37]	43.7	46.5	40.1	43.7	42.8	46.0	43.8
RTN [34]	75.3	97.1	98.3	66.9	85.6	85.7	84.8
JAN [17]	43.4	53.6	41.4	35.7	51.0	51.6	46.1
LEL [59]	73.2	93.9	96.8	76.4	83.6	84.8	84.8
IWAN [40]	89.2	99.3	99.4	90.5	**95.6**	84.7	93.1
SAN [39]	**93.9**	99.3	99.4	**94.3**	94.2	88.7	95.0
PADA [38]	86.5	99.3	**100.0**	82.2	92.7	95.4	92.7
DRCN (soft label)	88.5	**100.0**	**100.0**	86.0	**95.6**	**95.8**	94.3
DRCN (hard label)	90.8	**100.0**	**100.0**	**94.3**	95.2	94.8	**95.9**

4.3.2 Partial Transfer Tasks Results on Office-31

The classification results of Office-31 are reported in Table 3, from which we obviously observe that DRCN continuously outperforms other methods, however, by a small margin, e.g., average accuracy **94.3%** in soft label while **95.9%** in hard label, and SAN achieves the second highest accuracy of 95.0% followed by IWAN with 93.1%, while PADA gains 92.7%. It is clear that hard label based DRCN achieves comparable results while soft label based DRCN performs slightly worse, especially in task A→D where hard label based DRCN gets **8.3%** improvement compared to soft label based DRCN. This is because soft label based approach is likely to assign the weights to some unrelated classes, resulting in poor results, when classifier can accurately make prediction in a large probability. Additionally, we can get some interesting observations that DAN, RevGrad, ADDA and JAN all perform much worse than ResNet on most of tasks, by 6.4% to 55.4% descent, demonstrating that partial transfer task is a complex scenario which can not be easily tackled by traditional domain adaptation methods.

4.3.3 Partial Transfer Results on VisDA2017 and ImageNet-Caltech

In contrast to previous datasets, VisDA2017 and ImageNet-Caltech have much larger domain scales. However, DRCN can substantially improve the accuracy and is comparable to PADA, IWAN and SAN based on partial domain adaptation as shown in Table 4. Although PADA achieves the highest accuracy of 76.5% which is better than our best result of 74.2% on task **R**→**S**, but PADA is worse than DRCN on others. Especially for task **S**→**R**, DRCN significantly outperforms PADA by an increase of at least **3.7%**.

On the perspective of ImageNet-Caltech, our method promotes the accuracy in task **C**→**I** by a large margin, and gains comparable result in task **I**→**C**. We report the average accuracy on VisDA2017 and ImageNet-Caltech, manifesting that our method outperforms other baselines. Besides, soft label based DRCN has higher average accuracy than that of hard label based method, which shows that in a large-scale dataset, soft label has a greater impact in enhancing the contributions of relevant classes, and down-weighing contributions of irrelevant ones.

TABLE 4
Accuracy (%) on VisDA2017 and ImageNet-Caltech for **partial** transfer learning tasks (ResNet-50).

Method	VisDA2017		ImageNet-Caltech		Average
	R→S	S→R	I→C	C→I	
ResNet [45]	64.3	45.3	69.7	71.3	62.6
DAN [21]	68.4	47.6	71.3	60.1	61.9
RevGrad [18]	73.8	51.0	70.8	67.7	65.8
RTN [34]	72.9	50.0	75.5	66.2	66.2
IWAN [40]	71.3	48.6	**78.1**	73.3	67.8
SAN [39]	69.7	49.9	77.8	75.3	68.2
PADA [38]	**76.5**	53.5	75.0	70.5	68.9
DRCN (soft label)	73.2	**58.2**	75.3	**78.9**	**71.4**
DRCN (hard label)	74.2	57.2	74.1	77.5	70.8

Summary: First, from the results of Tables 2 and 3, we notice that traditional transfer learning methods cannot consistently outperform standard ResNet, which is extremely obvious in Table 3. However, some approaches on partial domain adaptation tasks can easily improve the accuracy by huge margins. This validates that partial transfer adaptation is a much more challenging problem for traditional domain adaptation methods. These methods only consider the general cross-domain features, which would lead to negative transfer.

Second, soft label based approach tends to work well in large-scale datasets (VisDA2017 and ImageNet-Caltech) while hard label based approach works well in small-scale datasets (Office-31 and Office-Home). This is because that classifier is prone to predict wrong label in large-scale dataset. In this case, hard label will falsely transfer unrequired class information, whereas soft label brings additional information from other classes which may cover desirable knowledge and then mitigates prediction deviations. However, if classifier works on small-scale datasets and produces correct prediction, it is detrimental to assign weights on those irrelevant classes. This problem can be avoided by hard label as its one-hot encoding attribute.

Third, from small-scale to large-scale datasets, DRCN achieves significant improvements on most tasks. Those encouraging results indicate that DRCN can effectively identify the most relevant source classes and learn more transferable features for partial domain adaptation.

4.4 Extension: Traditional Transfer Learning Tasks

As stated in Section 3, by enlarging the value of α and setting the class-wise weights equal to 1, i.e. $\boldsymbol{w} = \mathbf{1}$, DRCN can be easily generalized to deal with traditional domain adaptation scenarios. To evaluate the generalization ability, we extensively compare DRCN with several popular traditional domain adaptation methods including DAN [21], RevGrad [18], RTN [34], JAN [17], ADDA [37], MADA [35], CDAN [60], with two datasets on traditional transfer tasks, where the source domain and target domain have identical label spaces. For traditional transfer tasks, we set $\alpha = 1.5$ and $\beta = 0.05$ as default. The results are shown in Tables 5 and 6. Additionally, in the traditional domain adaptation scenario, DRCN could also be used to address the problems of the fine-grained recognition in the wild.

4.4.1 Traditional Transfer Results on Office-Home

The results on the 12 transfer tasks of Office-Home are shown in Table 5. We observe that DRCN outperforms all other methods on most tasks. Regarding those baseline methods, their average accuracy can be over 55% which is much better than the baseline ResNet. Additionally, CDAN+E wins the highest accuracies among baseline methods, with 7.5% and 2% improvements compared to JAN and CDAN respectively. For the average accuracy, we can observe that soft label based DRCN is better with hard label based DRCN on most tasks, and both of them can be competitive with CDAN+E. For example, on the task **Cl**→**Ar**, our method has increased by **4.3%** and **4.1%** respectively under soft label approach and hard label approach compared to CDAN+E. This suggests that our method can be successfully extended to address traditional transfer tasks.

4.4.2 Traditional Transfer Tasks Results on Office-31

Table 6 shows the classification accuracy results on Office-31. It is clearly that TCA and GFK are comparable to each other, while RevGrad and ADDA take slightly advantages over DAN and RTN. Similarly, MADA beats JAN by an increase of 0.9%. The average classification accuracy of DRCN of soft label on all 6 transfer tasks is **87.2%** and that of hard label on these tasks is **87.0%**, in which we can see that the best accuracy increase is 0.7% against CDAN. Although our approach is slightly worse than the best baseline CDAN+E by a marginal decrease of 0.5%, DRCN still can achieve accuracy improvements on some hard tasks, such as **D**→**A** and **W**→**A**, and attain comparable results on other easy tasks, indicating that DRCN also has potential power to substantially improve the classification accuracies in traditional domain adaptation scenarios.

Summary: First, as it can be seen in Table 5 and 6, DRCN (soft label) is slightly better than DRCN (hard label), which means the target prediction distribution could provide more discriminative knowledge comparing to one-hot pseudo target labels. Thus, it results in better performance when encountering traditional domain adaptation scenarios.

Second, by adjusting the values of α and setting $\boldsymbol{w} = \mathbf{1}$, we can get comparable or even better results against other

TABLE 5
Accuracy (%) on Office-Home for **traditional** transfer learning tasks (ResNet-50).

Method	Office-Home												
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
ResNet [45]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [21]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
RevGrad [18]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [17]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [60]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E [60]	**50.7**	70.6	76.0	57.6	**70.0**	70.0	57.4	**50.9**	**77.3**	70.9	**56.7**	**81.6**	65.8
DRCN (soft label)	50.6	**72.4**	**76.8**	**61.9**	69.5	**71.3**	**60.4**	48.6	76.8	72.9	56.1	81.4	**66.6**
DRCN (hard label)	49.1	71.2	76.1	61.7	**70.0**	71.1	59.4	49.1	76.8	**73.3**	55.9	80.8	66.2

Fig. 4. Examples of Website (Web) and Google Street View (GSV) Images for one type of car in the used fine-grained car dataset.

competitive domain adaptation methods. Those convincing results on challenging tasks show that the modified DRCN can also adapt to traditional domain adaptation problem.

Third, comparing with RTN [34], DRCN could achieve much higher prediction accuracies, which indicates mitigating the feature discrepancy between domains rather than classifier difference is more crucial for addressing domain adaptation problems.

TABLE 6
Accuracy (%) on Office-31 for **traditioanl** transfer learning tasks (ResNet-50).

Method	Office-31						
	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet [45]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
TCA [16]	72.7	96.7	99.6	74.1	61.7	60.9	77.6
GFK [58]	72.8	95.0	98.2	74.5	63.4	61.0	77.5
DAN [21]	80.5	97.1	99.6	78.6	63.6	62.8	80.4
RTN [34]	84.5	96.8	99.4	77.5	66.2	64.8	81.6
RevGrad [18]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA [37]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN [17]	85.4	97.4	99.8	84.7	68.6	70.0	84.3
MADA [35]	90.0	97.4	99.6	87.8	70.3	66.4	85.2
CDAN [60]	93.1	98.2	**100.0**	89.8	70.1	68.0	86.5
CDAN+E [60]	**94.1**	**98.6**	**100.0**	**92.9**	71.0	69.3	**87.7**
DRCN (soft label)	93.1	98.0	**100.0**	89.4	71.4	**71.0**	87.2
DRCN (hard label)	92.7	97.6	99.8	88.4	**72.7**	**71.0**	87.0

4.4.3 *Cross-domain Fine-grained Recognition in the Wild*

Fine-grained visual recognition is a well-studied problem to distinguish between objects in the same category (e.g. different car brands) [6], [7], [22], [61], [62]. However, it is infeasible to annotate enough data for every new scenario, thus addressing the problem of cross-domain fine-grained object recognition in the wild is more challenging and practical in computer vision.

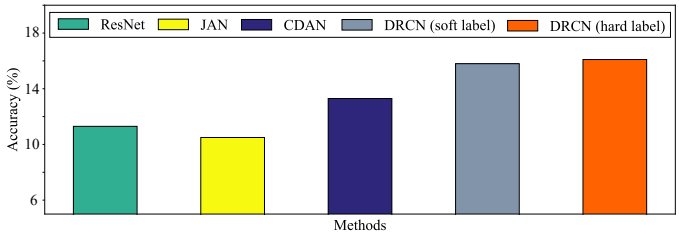


Fig. 5. Classification accuracies of ResNet, JAN, CDAN and DRCN on the task **Web** → **GSV** of fine-grained recognition in the wild.

Similar to [62], we can adapt the discriminative knowledge of annotated sources such as e-commerce websites to a sparse set of annotations in the real world, i.e. Google Street View. Fig. 4 illustrates the examples of a fine-grained car dataset introduced in [61], [62], which totally contains 1,095,021 images with more than 2,000 classes. The car images from craigslist.com, cars.com and edmunds.com are referred to as **Web** images, which are with high resolution and typically un-occluded. By contrast, the car images from Google Street View are blurry and occluded, which are denoted as **GSV** images.

It is apparently to observe that, Web and GSV images vary a lot in viewpoint, occlusion and resolution, leading to a large distribution discrepancy between these two sets. To evaluate the effectiveness of DRCN on the problem of fine-grained recognition in the wild, we construct the adaptation task **Web** → **GSV**. Here, we also use ResNet-50 pre-trained on ImageNet as the basic network. The results are illustrated in Fig. 5. From the results, we can clearly obtain that DRCN outperforms other comparisons on this challenging fine-grained car recognition in the wild dataset, and gain $\mathbf{42.5}\%$ and $\mathbf{21.1}\%$ relative increases when compared to ResNet [45] and CDAN [60], with respect to the target prediction accuracy. This manifests DRCN can be well generalized to the problems of fine-grained car categorization in the wild, and significantly boosts the universality of DRCN.

4.5 Empirical Analysis

In this section, we will conduct several empirical experiments to verify the effectiveness of DRCN in detail.

4.5.1 *Feature Visualization*

To effectively present the feature correction process of DRCN, in Fig. 6 and Fig. 7, we visualize the t-SNE embeddings [63] of source and target corrected representations for **traditional** and **partial** domain adaptation scenarios (task: A → W in Office-31) at different learning iterations. From Fig.

TABLE 7
Accuracy (%) of DRCN, DRCN ($\alpha = 0$), DRCN ($\beta = 0$), DRCN ($\boldsymbol{w} = \mathbf{1}$) and DRCN (w/o RCB) on Office-Home for **partial** transfer learning tasks (ResNet-50).

Method	Office-Home												
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
IWAN	53.9	54.5	78.1	61.3	48.0	63.3	54.2	**52.0**	**81.3**	76.5	56.8	**82.9**	63.6
SAN	44.4	68.7	74.6	**67.5**	65.0	**77.8**	59.8	44.7	80.1	72.2	50.2	78.7	65.3
PADA	52.0	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.1
DRCN($\alpha = 0$)	46.0	65.8	81.0	49.0	51.2	60.2	59.1	33.8	78.4	71.4	45.8	76.5	59.9
DRCN($\beta = 0$)	49.0	68.4	79.5	56.1	56.0	64.3	61.0	43.5	74.3	66.4	50.7	73.2	61.9
DRCN($\boldsymbol{w} = \mathbf{1}$)	**54.2**	69.7	82.7	51.5	65.7	62.4	61.3	44.3	76.8	77.8	50.9	76.1	64.4
DRCN (w/o RCB)	53.2	73.3	80.9	59.5	58.7	68.0	64.6	48.7	78.7	73.6	56.6	79.4	66.3
DRCN (soft label)	**54.0**	**76.4**	**83.1**	62.1	**66.3**	71.0	**68.3**	49.8	80.4	**77.9**	**58.1**	79.9	**68.9**
DRCN (hard label)	51.6	75.8	82.0	62.9	64.1	72.9	67.4	50.0	81.0	76.4	57.7	79.3	68.4

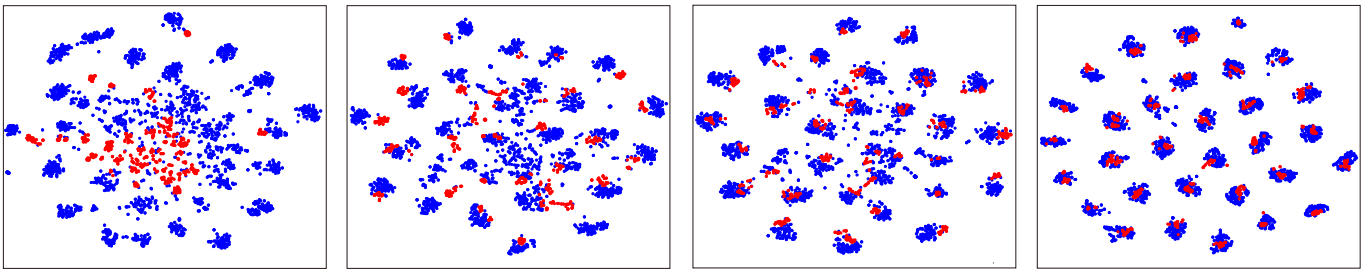


Fig. 6. t-SNE visualization of source and target corrected features for **traditional** domain adaptation in task A → W (31 classes) at different training iterations. Source data are blue dots and red dots represent target data.

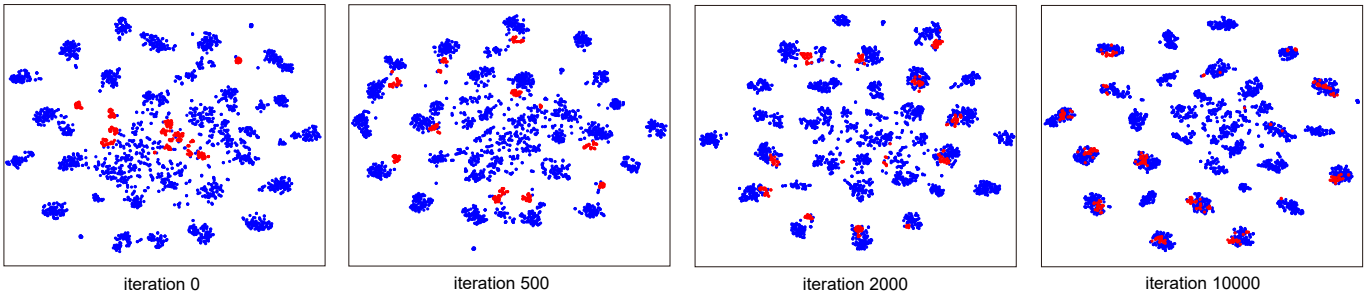


Fig. 7. t-SNE visualization of source and target corrected features for **partial** domain adaptation in task A → W (31 classes) at different training iterations. Source data are blue dots and red dots represent target data.

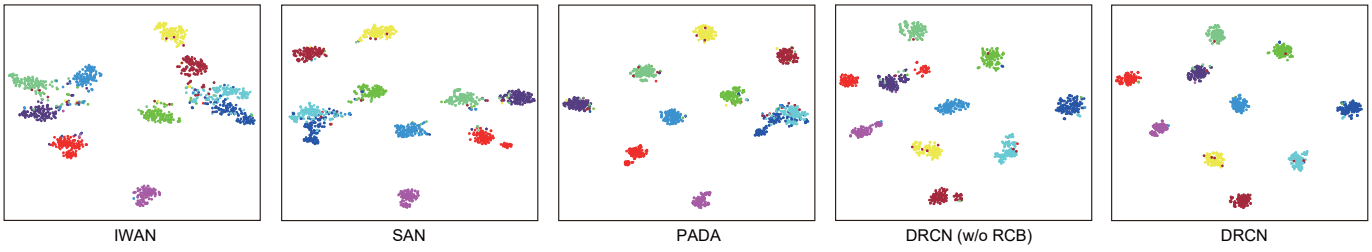


Fig. 8. t-SNE visualization of different methods (IWAN [40], SAN [39], PADA [38], DRCN w/o RCB, DRCN) for **partial** domain adaptation of task W → A (10 classes) in the shared class space. Each class is represented with different color.

6 and Fig. 7, we have the following observations. First, in the beginning, the source data are discriminated well while target data are substantially different with source, which demonstrates that there exists a large domain shift between source and target. Second, as the training process continues, the source and target representations become similar and each class across domains begins to align gradually. We attribute this phenomenon to the domain-wise and class-wise knowledge transfer and the powerful learning capacity of the plugged correction residual block. Third, by the end of training, source and target samples are nearly indistinguishable for both traditional and partial transfer scenarios, which manifests the effectiveness of DRCN to deal with domain adaptation problems.

Fig. 8 presents the t-SNE [63] visualization of the task-specific features learned by baselines, and of the features before and after the residual correction block (RCB) for DRCN in the shared class space. In IWAN and SAN, the data structures are more scattered, as there still has many outlier samples cannot converge into their class clusters. PADA can successfully mitigate domain shift. However, some categories are mixed. Different from them, DRCN can

not only successfully identify the most relevant classes but also represent good cohesion in clusters. These results verify the efficacy and superiority of DRCN comparing to other methods. Note that the feature learned by DRCN w/o RCB is not quite compact and some class centers are misaligned. However, they become more clustered after passing RCB in DRCN. This improvement benefits from the insertion of RCB, which further mitigates feature discrepancy.

4.5.2 Variants of DRCN

TABLE 8
Accuracy (%) of DRCN, DRCN ($\alpha = 0$), DRCN ($\beta = 0$), DRCN ($\boldsymbol{w} = \mathbf{1}$) and DRCN (w/o RCB) on Office-31 for **partial** transfer learning tasks (ResNet-50).

Method	Office-31						
	A→W	D→W	W→D	A→D	D→A	W→A	Average
IWAN	89.2	99.3	99.4	90.5	**95.6**	84.7	93.1
SAN	**93.9**	99.3	99.4	94.3	94.2	88.7	95.0
PADA	86.5	99.3	**100.0**	82.2	92.7	95.4	92.7
DRCN($\alpha = 0$)	89.0	99.3	**100.0**	93.6	89.1	93.3	94.1
DRCN($\beta = 0$)	86.4	**100.0**	**100.0**	84.7	93.4	92.5	93.3
DRCN($\boldsymbol{w} = \mathbf{1}$)	86.7	98.3	**100.0**	86.0	94.1	94.4	93.2
DRCN (w/o RCB)	86.8	**100.0**	99.6	86.6	93.4	93.4	93.3
DRCN (soft label)	88.5	**100.0**	**100.0**	86.0	**95.6**	**95.8**	94.3
DRCN (hard label)	90.8	**100.0**	**100.0**	**94.3**	95.2	94.8	**95.9**

We carry out four variants of DRCN (soft label) to go deeper into the influence of parameter α and β, the component residual correction block, as well as weights $\boldsymbol{w}$ on the classification performance. The results are reported in Tables 7 and 8. To be specific, DRCN ($\alpha = 0$) is the variant without loss of domain-wise knowledge transfer, i.e. without $\mathcal{L}_{domain}$. Similarly, DRCN ($\beta = 0$) is the variant without loss of class-wise knowledge transfer, while DRCN (w/o RCB) does not have residual correction block. DRCN ($\boldsymbol{w} = \mathbf{1}$) is the variant in which its each class weight $\boldsymbol{w}^{(k)}$ is 1 in Eq (9) with equal importance contribution for each class in terms of the class-wise alignment loss $\mathcal{L}_{class}$. Additionally, we test the performances of DRCN and its variants under **partial** transfer tasks on Office-Home and Office-31 datasets.

The results in Tables 7 and 8 reveal the following observations. First, it is obvious that DRCN (hard label), DRCN (soft label) and other variants of DRCN, except DRCN ($\alpha = 0$), all outperform PADA. This fact indicates that minimizing the domain-wise distribution difference is also important for tackling partial domain adaptation problems. Second, by comparing DRCN with DRCN ($\beta = 0$), we observe that DRCN outperforms DRCN ($\beta = 0$) with over 1% improvement in Office-31, even achieving 7 % in Office-Home, demonstrating that our designed weighted class-wise alignment is an essential part in DRCN. Third, we can find out that DRCN (hard label) and DRCN (soft label) overpass IWAN and SAN with a large margin, whereas DRCN ($\boldsymbol{w} = \mathbf{1}$) is worse than SAN. This is reasonable if we fix all class weight as 1, which would easily trigger negative transfer from irrelevant classes. Fourth, DRCN outperforms DRCN (w/o RCB) in both Office-31 and Office-Home. In Office-31, we can see over 1% accuracy decrease in DRCN (w/o RCB). The performance decline is more serious in Office-Home, with DRCN (w/o RCB) degrading more than 2%. Thus, the effectiveness of Residual Correction Block (RCB) can be successfully verified.

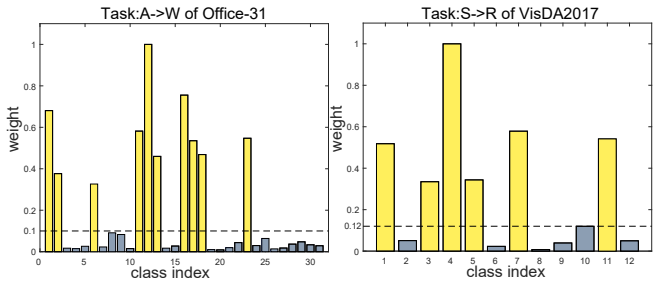


Fig. 9. Histograms of different class weights learned by DRCN for task: A → W of Office-31 and S → R of VisDA2017.

As a result, we can safely conclude that both class-wise knowledge and domain-wise knowledge need to be transferred simultaneously. Furthermore, residual correction block indeed helps enhance DRCN with greater capacity to knowledge transfer, and weights assigned to class is crucial to enabling positive transfer in the shared label space. Therefore, each component in DRCN is trying to solve partial domain adaptation from distinct aspect.

4.5.3 Statistics of Class Weights

For partial domain adaptation, to verify the effectiveness of the weighted scheme in DRCN, Fig. 9 presents the learned class weights for task: A → W of Office-31 and S → R of VisDA2017. The yellow bars represent the weights of shared classes, and grey bars are for irrelevant classes.

It is inspiring to see that the weights of shared classes are much larger than that of irrelevant classes. For task A → W as an example, all the weights of irrelevant classes are less than 0.1, and even the smallest weight of shared class is still several times of the largest weight of irrelevant classes. This weighted strategy by leveraging the target output distribution indeed can identify the most similar source subclasses to target domain, and improve the positive effects of them, simultaneously reduce the influence of irrelevant source data. Then, the weighted class-wise matching in DRCN will transfer the most relevant knowledge from a large-scale source domain to a small-scale target domain effectively.

4.5.4 Parameter Sensitivity

As shown in Fig. 10, we want to investigate the effects of parameters α and β on the experimental results by varying $\alpha \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$ and $\beta \in \{0, 0.02, 0.04, 0.05, 0.06, 0.08, 0.1\}$. Besides, we use two random transfer tasks which are **Ar→Cl** (Office-Home) and **A→D** (Office-31) to testify the performance.

Concerning with the parameter α, we observe that as α varies from 0 to 0.25, the classification accuracy of Office-31 decreases, while that of Office-Home increases. It is desirable that when Office-31 dataset has been aligned with the source and target domains, increasing α, in other words, preserving more general knowledge, will lead to negative transfer. However, the domains in Office-Home are very dissimilar with each other, only transferring task-level knowledge would not enough unless taking general knowledge into consideration. Therefore, for easy transfer tasks, we can safely choose a small α, but for hard transfer

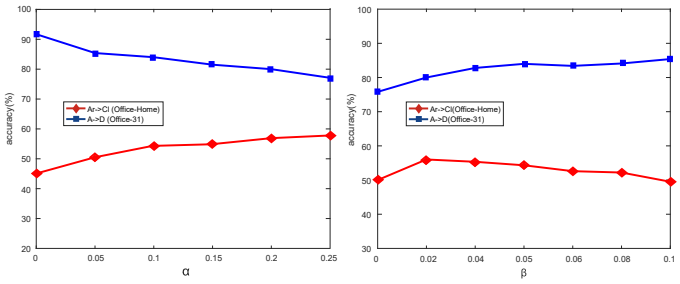


Fig. 10. Parameter sensitivity studies w.r.t. α and β respectively.

tasks, a relative large α will facilitate the general domain knowledge transfer.

As for the parameter β, the accuracy in Office-31 shows a growing trend with the increase of β, which further validates the importance of strengthening class-wise alignment under circumstances similar to Office-31. While the accuracy of Office-Home increases first and then decreases slightly as β increases. Note that there is a slight increase when $\beta = 0.02$, but the overall trend is still decreasing. An interpretation is that the average accuracy of Office-Home on the task **Ar→Cl** is very low, which indicates most of class-wise alignment can be incorrect. Therefore, increasing β means emphasizing the contribution of misalignment, and that will lead to more false classification results. Those observations demonstrate that proper trade-off will enhance effective knowledge transfer in DRCN.

Thus, if we choose reasonable values for α and β, DRCN could perform stably and outperform other competitors.

4.5.5 Layer Responses Analysis

As shown in Fig. 11 (a), to quantitatively characterize how much information the added residual block has learned, we calculate the mean and variance of the task-specific layer response $F_s(\boldsymbol{x}_t)$ and $F_s(\boldsymbol{x}_s)$, added residual correction layer response $\Delta F_s(\boldsymbol{x}_t)$ and their element-wise summation $F_t(\boldsymbol{x}_t)$ on task: W → A (Office-31), respectively.

The results show that the added residual block indeed has learned the discrepancy between source and target. Because the mean and variance values of $F_t(\boldsymbol{x}_t)$, which are calculated using features after residual correction block, are much similar with that of $F_s(\boldsymbol{x}_s)$, whereas there exists a relatively large gap between $F_s(\boldsymbol{x}_t)$ and $F_s(\boldsymbol{x}_s)$ in their values. Therefore, if the target data only passes through the original source network, the domain shift couldn't be mitigated so well without residual correction. This result manifests the added residual correction block has the powerful learning capacity for effective adaptation.

4.5.6 Proxy $\mathcal{A}$-distance for Traditional Domain Adaptation

$\mathcal{A}$-distance is a measure of distance between two distributions [64], and a larger $\mathcal{A}$-distance implies it is easier to discriminate source and target. Since directly computing $\mathcal{A}$-distance is intractable, we resort to leverage proxy $\mathcal{A}$-distance instead. The proxy $\mathcal{A}$-distance is defined as

$$\widehat{d}_{\mathcal{A}} = 2(1 - 2\epsilon), \tag{13}$$

where ϵ is the classification error of classifying source and target data with task-specific representations.

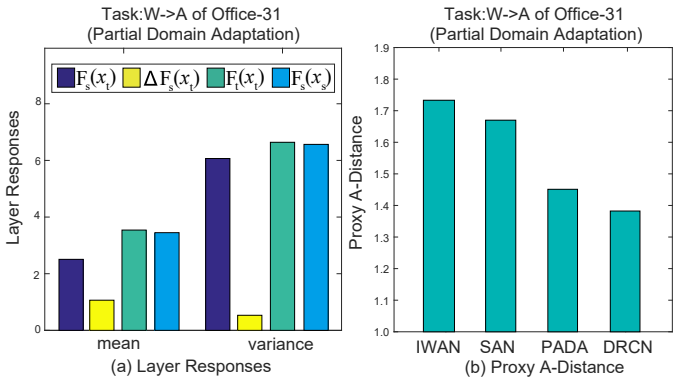


Fig. 11. (a) Statistics of task-specific layer response $F_s(\boldsymbol{x}_t)$ and $F_s(\boldsymbol{x}_s)$, added residual correction layer response $\Delta F_s(\boldsymbol{x}_t)$ and their element-wise summation $F_t(\boldsymbol{x}_t)$ on task: W → A of Office-31 (**partial** domain adaptation); (b) Proxy $\mathcal{A}$-distance of IWAN [40], SAN [39], PADA [38] and DRCN features on task: W → A of Office-31 (**partial** domain adaptation).

To evaluate the effectiveness of learned features for different methods, we calculate the proxy $\mathcal{A}$-distance of IWAN [40], SAN [39], PADA [38] features extracted after the task-specific layer, and DRCN features $F_s(\boldsymbol{x}_s), F_t(\boldsymbol{x}_t)$ for task: W → A of Office-31 (partial domain adaptation scenario). The results are shown in Fig. 11(b).

We observe that DRCN has the smallest $\widehat{d}_{\mathcal{A}}$ among the four methods, followed by PADA, SAN, and IWAN. It is known that the smaller the A-distance is, the easier the feature is to transfer. The $\widehat{d}_{\mathcal{A}}$ of PADA features is less than IWAN and SAN, which implies PADA features could learn transferable representations to confuse classifier. However, the $\widehat{d}_{\mathcal{A}}$ using DRCN features is smaller than that of PADA. We therefore can argue that DRCN features are more indistinguishable and can close the domain shift effectively, which explains the better performance of DRCN than competitive methods.

5 Conclusion

This paper introduces a novel Deep Residual Correction Network (DRCN) to address partial domain adaptation problems, in which the target label space is a subset of source label space. DRCN could learn the different significance of source classes automatically by leveraging the target output probability distribution. Based on the learned weights, we propose a weighted class-wise matching strategy to explicitly align target data with the most relevant source subclasses, and maximally mitigate the discrepancy across domains. Different from other partial domain adaptation architectures, DRCN also jointly transfers general feature-level and task-level knowledge from source to target, since we find that properly transferring general knowledge can benefit the final classification significantly as well. To boost the adaptation ability of structure, DRCN plugs one residual correction block into the general source network along with the task-specific feature layer, which is easily implemented and efficiently generalized to new designed deep networks. DRCN can be easily extended to deal with traditional and fine-grained cross-domain visual recognition tasks. Extensive experimental results on several standard cross-domain datasets have demonstrated that

DRCN outperforms several competitive deep domain adaptation approaches in both partial and traditional domain adaptation scenarios by a large margin.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[2] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, "Interleaved group convolutions," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2017.

[3] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2017, pp. 5987–5995.

[4] G. Xie, J. Wang, T. Zhang, J. Lai, R. Hong, and G.-J. Qi, "Interleaved structured sparse convolutional neural networks," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2018, pp. 8847–8856.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[6] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Int. Conf. Comput. Vis. (ICCV)*, vol. 6, 2017.

[7] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, vol. 2, 2017, p. 3.

[8] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[9] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3104–3112.

[12] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv:1705.03122*, 2017.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng. (TKDE)*, vol. 22, no. 10, pp. 1345–1359, 2010.

[14] J. Liang, R. He, Z. Sun, and T. Tan, "Aggregating randomized clustering-promoting invariant projections for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2018.

[15] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv:1702.05374*, 2017.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw. (TNN)*, vol. 22, no. 2, pp. 199–210, 2011.

[17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2208–2217.

[18] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1180–1189.

[19] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 2456–2464.

[20] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Int. Conf. Mach. Learn. (ICML)*. ACM, July 2012, pp. 767–774.

[21] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 97–105.

[22] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2018, pp. 4109–4118.

[23] A. K. Rajagopal, R. Subramanian, E. Ricci, R. L. Vieriu, O. Lanz, N. Sebe *et al.*, "Exploring transfer learning approaches for head pose classification from multi-view surveillance images," *Int. J. of Comput. Vis. (IJCV)*, vol. 109, no. 1-2, pp. 146–167, 2014.

[24] Y. Yan, E. Ricci, R. Subramanian, G. Liu, O. Lanz, and N. Sebe, "A multi-task learning framework for head pose estimation under target motion," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 38, no. 6, pp. 1070–1083, 2016.

[25] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 601–608.

[26] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Sys. (TNNLS)*, vol. 28, no. 7, pp. 1682–1695, 2017.

[27] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Int. Conf. Comput. Vis. (ICCV)*. IEEE, 2013, pp. 2200–2207.

[28] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2017, pp. 5150–5158.

[29] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE Trans. Image Process. (TIP)*, vol. 27, no. 9, pp. 4260–4273, 2018.

[30] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition." in *Int. Conf. Mach. Learn. (ICML)*.

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3320–3328.

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

[33] S. Li, S.-j. Song, and C. Wu, "Layer-wise domain correction for unsupervised domain adaptation," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 91–103, 2018.

[34] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 136–144.

[35] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI Conf. Art. Intell. (AAAI)*, 2018.

[36] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4068–4076.

[37] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 4.

[38] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 135–150.

[39] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2018, pp. 2724–2732.

[40] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2018, pp. 8156–8164.

[41] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res. (JMLR)*, vol. 12, pp. 2493–2537, 2011.

[42] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI Conf. Art. Intell. (AAAI)*, 2016, pp. 2058–2065.

[43] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, no. 1, pp. 1–1, 2017.

[44] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

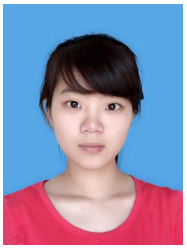
[46] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 343–351.

[47] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 7.

[48] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 513–520.

[49] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.
[50] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1718–1727.
[51] J. R. Gardner, P. Upchurch, M. J. Kusner, Y. Li, K. Q. Weinberger, K. Bala, and J. E. Hopcroft, "Deep manifold traversal: Changing labels with convolutional features," *arXiv:1511.06421*, 2015.
[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.
[53] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Int. Conf. Mach. Learn. (ICML)*. ACM, 2009, pp. 961–968.
[54] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, 2017, pp. 5018–5027.
[55] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 213–226.
[56] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2018.
[57] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.
[58] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2066–2073.
[59] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei, "Label efficient learning of transferable representations acrosss domains and tasks," in *Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 165–177.
[60] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1640–1650.
[61] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, and L. Fei-Fei, "Fine-grained car detection for visual census estimation," in *AAAI Conf. Art. Intell. (AAAI)*, 2017.
[62] T. Gebru, J. Hoffman, and L. Fei-Fei, "Fine-grained recognition in the wild: A multi-task domain adaptation approach," in *Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1349–1358.
[63] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res. (JMLR)*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[64] S. Bendavid, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn. (MLJ)*, vol. 79, no. 1-2, pp. 151–175, 2010.

Chi Harold Liu (SM'15) receives the Ph.D. degree from Imperial College, UK in 2010, and the he B.Eng. de. degree from Tsinghua University, China in 2006. He is currently a Full Professor and Vice Dean at the School of Computer Science and Technology, Beijing Institute of Technology, China. He is also the Director of IBM Mainframe Excellence Center (Beijing), Director of IBM Big Data Technology Center, and Director of National Laboratory of Data Intelligence for China Light Industry. Before moving to academia, he joined IBM Research - China as a staff researcher and project manager, after working as a postdoctoral researcher at Deutsche Telekom Laboratories, Germany, and a visiting scholar at IBM T. J. Watson Research Center, USA. His current research interests include the Internet-of-Things (IoT), Big Data analytics, mobile computing, and deep learning. He received the Distinguished Young Scholar Award in 2013, IBM First Plateau Invention Achievement Award in 2012, and IBM First Patent Application Award in 2011 and was interviewed by EEWeb.com as the Featured Engineer in 2011. He has published more than 80 prestigious conference and journal papers and owned more than 14 EU/U.S./U.K./China patents. He serves as the Area Editor for KSII Trans. on Internet and Information Systems and the book editor for six books published by Taylor & Francis Group, USA and China Machinery Press. He also has served as the general chair of IEEE SECON'13 workshop on IoT Networking and Control, IEEE WCNC'12 workshop on IoT Enabling Technologies, and ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He served as the consultant to Asian Development Bank, Bain & Company, and KPMG, USA, and the peer reviewer for Qatar National Research Foundation, and National Science Foundation, China. He is a Senior Member of IEEE.

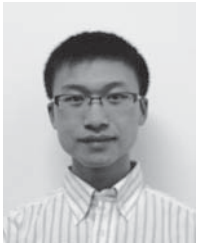


Qiuxia Lin is pursuing the M.S. degree in Computer Science from Beijing Institute of Technology. Her research interests include deep learning and transfer learning.

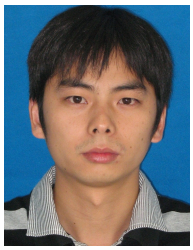
Qi Wen is pursuing the M.E. degree in Computer Science from Beijing Institute of Technology. His research interests include deep learning and transfer learning.

Shuang Li is an assistant professor in the school of Computer Science and Technology, Beijing Institute of Technology. He received his Ph.D. degree in Control Science and Engineering from Department of Automation, Tsinghua University, China in 2018. He was a visiting research scholar at the Department of Computer Science, Cornell University from November 2015 to June 2016. His main research interests include machine learning and deep learning, especially in transfer learning & domain adaptation.

Limin Su is pursuing the B.E. degree in Computer Science from Beijing Institute of Technology. Her research interests include machine learning and transfer learning.



Gao Huang is an assistant professor in the Department of Automation, Tsinghua University. He was a Postdoctoral Researcher in the Department of Computer Science at Cornell University. He received the PhD degree in Control Science and Engineering from Tsinghua University in 2015, and B.Eng degree in Automation from Beihang University in 2009. He was a visiting student at Washington University at St. Louis and Nanyang Technological University in 2013 and 2014, respectively. His research interests include machine learning and computer vision.



Zhengming Ding (S'14) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from University of Electronic Science and Technology of China (UESTC), China, in 2010 and 2013, respectively. He received the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, USA in 2018. He is a faculty member affiliated with Department of Computer, Information and Technology, Indiana University-Purdue University Indianapolis since 2018. His research interests include machine learning and computer vision. Specifically, he devotes himself to develop scalable algorithms for challenging problems in transfer learning and deep learning scenario. He received the National Institute of Justice Fellowship during 2016-2018. He was the recipients of the best paper award (SPIE 2016) and best paper candidate (ACM MM 2017). He is now an Associate Editor for the Journal of Electronic Imaging (JEI).