Deep Transfer Learning with Joint Adaptation Networks

Mingsheng Long [1] **Han Zhu** [1] **Jianmin Wang** [1] **Michael I. Jordan** [2]

Abstract

Deep networks have been successfully applied to learn transferable features for adapting models from a source domain to a different target domain. In this paper, we present joint adaptation networks (JAN), which learn a transfer network by aligning the joint distributions of multiple domain-specific layers across domains based on a joint maximum mean discrepancy (JMMD) criterion. Adversarial training strategy is adopted to maximize JMMD such that the distributions of the source and target domains are made more distinguishable. Learning can be performed by stochastic gradient descent with the gradients computed by back-propagation in linear-time. Experiments testify that our model yields state of the art results on standard datasets.

1. Introduction

Deep networks have significantly improved the state of the arts for diverse machine learning problems and applications. Unfortunately, the impressive performance gains come only when massive amounts of labeled data are available for supervised learning. Since manual labeling of sufficient training data for diverse application domains on-the-fly is often prohibitive, for a target task short of labeled data, there is strong motivation to build effective learners that can leverage rich labeled data from a different source domain. However, this learning paradigm suffers from the shift in data distributions across different domains, which poses a major obstacle in adapting predictive models for the target task (Quionero-Candela et al., 2009; Pan & Yang, 2010).

Learning a discriminative model in the presence of the shift between training and test distributions is known as transfer learning or domain adaptation (Pan & Yang, 2010). Previous shallow transfer learning methods bridge the source and target domains by learning invariant feature representations or estimating instance importance without using target labels (Huang et al., 2006; Pan et al., 2011; Gong et al., 2013). Recent deep transfer learning methods leverage deep networks to learn more transferable representations by embedding domain adaptation in the pipeline of deep learning, which can simultaneously disentangle the explanatory factors of variations behind data and match the marginal distributions across domains (Tzeng et al., 2014; 2015; Long et al., 2015; 2016; Ganin & Lempitsky, 2015; Bousmalis et al., 2016).

Transfer learning becomes more challenging when domains may change by the joint distributions of input features and output labels, which is a common scenario in practical applications. First, deep networks generally learn the complex function from input features to output labels via multilayer feature transformation and abstraction. Second, deep features in standard CNNs eventually transition from general to specific along the network, and the transferability of features and classifiers decreases when the cross-domain discrepancy increases (Yosinski et al., 2014). Consequently, after feed-forwarding the source and target domain data through deep networks for multilayer feature abstraction, the shifts in the joint distributions of input features and output labels still linger in the network activations of multiple domain-specific higher layers. Thus we can use the joint distributions of the activations in these domain-specific layers to approximately reason about the original joint distributions, which should be matched across domains to enable domain adaptation. To date, this problem has not been addressed in deep networks.

In this paper, we present Joint Adaptation Networks (JAN) to align the joint distributions of multiple domain-specific layers across domains for unsupervised domain adaptation. JAN largely extends the ability of deep adaptation networks (Long et al., 2015) to reason about the joint distributions as mentioned above, while keeping the training procedure even simpler. Specifically, JAN admits a simple transfer pipeline, which processes the source and target domain data by convolutional neural networks (CNN) and then aligns the joint distributions of activations in multiple task-specific layers. To learn parameters and enable alignment, we derive joint maximum mean discrepancy (JMMD), which measures the Hilbert-Schmidt norm between kernel mean embedding of empirical joint distributions of source and target data.

[1]Key Lab for Information System Security, MOE; Tsinghua National Lab for Information Science and Technology (TNList); NEL-BDS; School of Software, Tsinghua University, Beijing 100084, China [2]University of California, Berkeley, Berkeley 94720. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

Thanks to a linear-time unbiased estimate of JMMD, we can easily draw a mini-batch of samples to estimate the JMMD criterion, and implement it efficiently via back-propagation. We further maximize JMMD using adversarial training strategy such that the distributions of source and target domains are made more distinguishable. Empirical study shows that our models yield state of the art results on standard datasets.

2. Related Work

Transfer learning (Pan & Yang, 2010) aims to build learning machines that generalize across different domains following different probability distributions (Sugiyama et al., 2008; Pan et al., 2011; Duan et al., 2012; Gong et al., 2013; Zhang et al., 2013). Transfer learning finds wide applications in computer vision (Saenko et al., 2010; Gopalan et al., 2011; Gong et al., 2012; Hoffman et al., 2014) and natural language processing (Collobert et al., 2011; Glorot et al., 2011).

The main technical problem of transfer learning is how to reduce the shifts in data distributions across domains. Most existing methods learn a shallow representation model by which domain discrepancy is minimized, which cannot suppress domain-specific exploratory factors of variations. Deep networks learn abstract representations that disentangle the explanatory factors of variations behind data (Bengio et al., 2013) and extract transferable factors underlying different populations (Glorot et al., 2011; Oquab et al., 2013), which can only reduce, but not remove, the cross-domain discrepancy (Yosinski et al., 2014). Recent work on deep domain adaptation embeds domain-adaptation modules into deep networks to boost transfer performance (Tzeng et al., 2014; 2015; 2017; Ganin & Lempitsky, 2015; Long et al., 2015; 2016). These methods mainly correct the shifts in marginal distributions, assuming conditional distributions remain unchanged after the marginal distribution adaptation.

Transfer learning will become more challenging as domains may change by the joint distributions $P(\mathbf{X}, \mathbf{Y})$ of input features $\mathbf{X}$ and output labels $\mathbf{Y}$. The distribution shifts may stem from the marginal distributions $P(\mathbf{X})$ (a.k.a. covariate shift (Huang et al., 2006; Sugiyama et al., 2008)), the conditional distributions $P(\mathbf{Y}|\mathbf{X})$ (a.k.a. conditional shift (Zhang et al., 2013)), or both (a.k.a. dataset shift (Quionero-Candela et al., 2009)). Another line of work (Zhang et al., 2013; Wang & Schneider, 2014) correct both target and conditional shifts based on the theory of kernel embedding of conditional distributions (Song et al., 2009; 2010; Sriperumbudur et al., 2010). Since the target labels are unavailable, adaptation is performed by minimizing the discrepancy between marginal distributions instead of conditional distributions. In general, the presence of conditional shift leads to an ill-posed problem, and an additional assumption that the conditional distribution may only change under location-scale transformations on $\mathbf{X}$ is commonly imposed to make the problem tractable (Zhang et al., 2013). As it is not easy to justify which components of the joint distribution are changing in practice, our work is transparent to diverse scenarios by directly manipulating the joint distribution without assumptions on the marginal and conditional distributions. Furthermore, it remains unclear how to account for the shift in joint distributions within the regime of deep architectures.

3. Preliminary

3.1. Hilbert Space Embedding

We begin by providing an overview of Hilbert space embeddings of distributions, where each distribution is represented by an element in a reproducing kernel Hilbert space (RKHS). Denote by $\mathbf{X}$ a random variable with domain Ω and distribution $P(\mathbf{X})$, and by $\mathbf{x}$ the instantiations of $\mathbf{X}$. A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ on Ω endowed by a kernel $k(\mathbf{x}, \mathbf{x}')$ is a Hilbert space of functions $f : \Omega \mapsto \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Its element $k(\mathbf{x}, \cdot)$ satisfies the reproducing property: $\langle f(\cdot), k(\mathbf{x}, \cdot) \rangle_{\mathcal{H}} = f(\mathbf{x})$. Alternatively, $k(\mathbf{x}, \cdot)$ can be viewed as an (infinite-dimensional) implicit feature map $\phi(\mathbf{x})$ where $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$. Kernel functions can be defined on vector space, graphs, time series and structured objects to handle diverse applications. The kernel embedding represents a probability distribution P by an element in RKHS endowed by a kernel k (Smola et al., 2007; Sriperumbudur et al., 2010; Gretton et al., 2012)

$$\mu_{\mathbf{X}}(P) \triangleq \mathbb{E}_{\mathbf{X}}[\phi(\mathbf{X})] = \int_{\Omega} \phi(\mathbf{x}) \, \mathrm{d}P(\mathbf{x}), \tag{1}$$

where the distribution is mapped to the expected feature map, i.e. to a point in the RKHS, given that $\mathbb{E}_{\mathbf{X}}[k(\mathbf{x}, \mathbf{x}')] \leqslant \infty$. The mean embedding $\mu_{\mathbf{X}}$ has the property that the expectation of any RKHS function f can be evaluated as an inner product in $\mathcal{H}$, $\langle \mu_{\mathbf{X}}, f \rangle_{\mathcal{H}} \triangleq \mathbb{E}_{\mathbf{X}}[f(\mathbf{X})], \forall f \in \mathcal{H}$. This kind of kernel mean embedding provides us a nonparametric perspective on manipulating distributions by drawing samples from them. We will require a characteristic kernel k such that the kernel embedding $\mu_{\mathbf{X}}(P)$ is injective, and that the embedding of distributions into infinite-dimensional feature spaces can preserve all of the statistical features of arbitrary distributions, which removes the necessity of density estimation of P. This technique has been widely applied in many tasks, including feature extraction, density estimation and two-sample test (Smola et al., 2007; Gretton et al., 2012).

While the true distribution $P(\mathbf{X})$ is rarely accessible, we can estimate its embedding using a finite sample (Gretton et al., 2012). Given a sample $\mathcal{D}_{\mathbf{X}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of size n drawn i.i.d. from $P(\mathbf{X})$, the empirical kernel embedding is

$$\widehat{\mu}_{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i). \tag{2}$$

This empirical estimate converges to its population counterpart in RKHS norm $\|\mu_{\mathbf{X}} - \widehat{\mu}_{\mathbf{X}}\|_{\mathcal{H}}$ with a rate of $O(n^{-\frac{1}{2}})$.

Kernel embeddings can be readily generalized to *joint* distributions of two or more variables using tensor product feature spaces (Song et al., 2009; 2010; Song & Dai, 2013). A joint distribution P of variables $\mathbf{X}^1, \ldots, \mathbf{X}^m$ can be embedded into an m-th order tensor product feature space $\otimes_{\ell=1}^m \mathcal{H}^\ell$ by

$$\begin{aligned}\mathcal{C}_{\mathbf{X}^{1:m}}(P) &\triangleq \mathbb{E}_{\mathbf{X}^{1:m}}\left[\otimes_{\ell=1}^m \phi^\ell\left(\mathbf{X}^\ell\right)\right] \\ &= \int_{\times_{\ell=1}^m \Omega^\ell} \left(\otimes_{\ell=1}^m \phi^\ell\left(\mathbf{x}^\ell\right)\right) \mathrm{d}P\left(\mathbf{x}^1, \ldots, \mathbf{x}^m\right),\end{aligned} \tag{3}$$

where $\mathbf{X}^{1:m}$ denotes the set of m variables $\{\mathbf{X}^1, \ldots, \mathbf{X}^m\}$ on domain $\times_{\ell=1}^m \Omega^\ell = \Omega^1 \times \ldots \times \Omega^m$, ϕ^ℓ is the feature map endowed with kernel k^ℓ in RKHS $\mathcal{H}^\ell$ for variable $\mathbf{X}^\ell$, $\otimes_{\ell=1}^m \phi^\ell\left(\mathbf{x}^\ell\right) = \phi^1\left(\mathbf{x}^1\right) \otimes \ldots \otimes \phi^m\left(\mathbf{x}^m\right)$ is the feature map in the tensor product Hilbert space, where the inner product satisfies $\langle \otimes_{\ell=1}^m \phi^\ell(\mathbf{x}^\ell), \otimes_{\ell=1}^m \phi^\ell(\mathbf{x}'^\ell)\rangle = \prod_{\ell=1}^m k^\ell(\mathbf{x}^\ell, \mathbf{x}'^\ell)$. The joint embeddings can be viewed as an uncentered cross-covariance operator $\mathcal{C}_{\mathbf{X}^{1:m}}$ by the standard equivalence between tensor and linear map (Song et al., 2010). That is, given a set of functions $f^1, \ldots, f^m$, their covariance can be computed by $\mathbb{E}_{\mathbf{X}^{1:m}}\left[\prod_{\ell=1}^m f^\ell(\mathbf{X}^\ell)\right] = \left\langle \otimes_{\ell=1}^m f^\ell, \mathcal{C}_{\mathbf{X}^{1:m}}\right\rangle$.

When the true distribution $P(\mathbf{X}^1, \ldots, \mathbf{X}^m)$ is unknown, we can estimate its embedding using a finite sample (Song et al., 2013). Given a sample $\mathcal{D}_{\mathbf{X}^{1:m}} = \{\mathbf{x}_1^{1:m}, \ldots, \mathbf{x}_n^{1:m}\}$ of size n drawn i.i.d. from $P(\mathbf{X}^1, \ldots, \mathbf{X}^m)$, the empirical joint embedding (the cross-covariance operator) is estimated as

$$\widehat{\mathcal{C}}_{\mathbf{X}^{1:m}} = \frac{1}{n} \sum_{i=1}^n \otimes_{\ell=1}^m \phi^\ell\left(\mathbf{x}_i^\ell\right). \tag{4}$$

This empirical estimate converges to its population counterpart with a similar convergence rate as marginal embedding.

3.2. Maximum Mean Discrepancy

Let $\mathcal{D}_{\mathbf{X}^s} = \{\mathbf{x}_1^s, \ldots, \mathbf{x}_{n_s}^s\}$ and $\mathcal{D}_{\mathbf{X}^t} = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_{n_t}^t\}$ be the sets of samples from distributions $P(\mathbf{X}^s)$ and $Q(\mathbf{X}^t)$, respectively. Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) is a kernel two-sample test which rejects or accepts the null hypothesis $P = Q$ based on the observed samples. The basic idea behind MMD is that if the generating distributions are identical, all the statistics are the same. Formally, MMD defines the following difference measure:

$$D_{\mathcal{H}}(P, Q) \triangleq \sup_{f \in \mathcal{H}} \left(\mathbb{E}_{\mathbf{X}^s}\left[f\left(\mathbf{X}^s\right)\right] - \mathbb{E}_{\mathbf{X}^t}\left[f\left(\mathbf{X}^t\right)\right]\right), \tag{5}$$

where $\mathcal{H}$ is a class of functions. It is shown that the class of functions in an universal RKHS $\mathcal{H}$ is rich enough to distinguish any two distributions and MMD is expressed as the distance between their mean embeddings: $D_{\mathcal{H}}(P, Q) = \|\mu_{\mathbf{X}^s}(P) - \mu_{\mathbf{X}^t}(Q)\|_{\mathcal{H}}^2$. The main theoretical result is that $P = Q$ if and only if $D_{\mathcal{H}}(P, Q) = 0$ (Gretton et al., 2012). In practice, an estimate of the MMD compares the square distance between the empirical kernel mean embeddings as

$$\begin{aligned}\widehat{D}_{\mathcal{H}}(P, Q) = &\frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k\left(\mathbf{x}_i^s, \mathbf{x}_j^s\right) \\ &+ \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k\left(\mathbf{x}_i^t, \mathbf{x}_j^t\right) \\ &- \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k\left(\mathbf{x}_i^s, \mathbf{x}_j^t\right),\end{aligned} \tag{6}$$

where $\widehat{D}_{\mathcal{H}}(P, Q)$ is an unbiased estimator of $D_{\mathcal{H}}(P, Q)$.

4. Joint Adaptation Networks

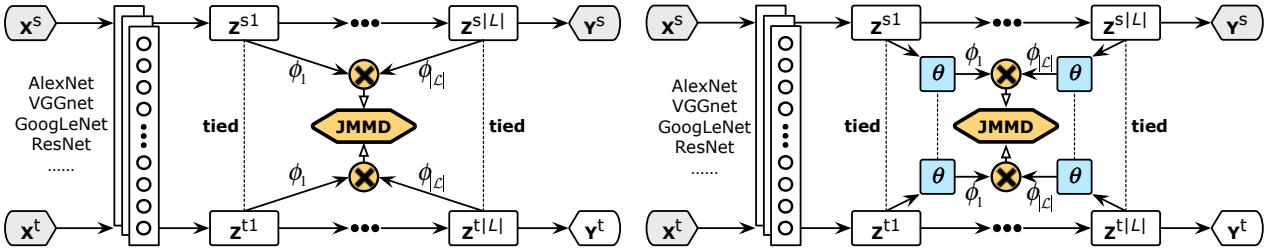
In unsupervised domain adaptation, we are given a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples. The source domain and target domain are sampled from joint distributions $P(\mathbf{X}^s, \mathbf{Y}^s)$ and $Q(\mathbf{X}^t, \mathbf{Y}^t)$ respectively, $P \neq Q$. The goal of this paper is to design a deep neural network $\mathbf{y} = f(\mathbf{x})$ which formally reduces the shifts in the joint distributions across domains and enables learning both transferable features and classifiers, such that the target risk $R_t(f) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim Q}\left[f(\mathbf{x}) \neq \mathbf{y}\right]$ can be minimized by jointly minimizing the source risk and domain discrepancy.

Recent studies reveal that deep networks (Bengio et al., 2013) can learn more transferable representations than traditional hand-crafted features (Oquab et al., 2013; Yosinski et al., 2014). The favorable transferability of deep features leads to several state of the art deep transfer learning methods (Ganin & Lempitsky, 2015; Tzeng et al., 2015; Long et al., 2015; 2016). This paper also tackles unsupervised domain adaptation by learning transferable features using deep neural networks. We extend deep convolutional neural networks (CNNs), including AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016), to novel joint adaptation networks (JANs) as shown in Figure 1. The empirical error of CNN classifier $f(\mathbf{x})$ on source domain labeled data $\mathcal{D}_s$ is

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right), \tag{7}$$

where $J(\cdot, \cdot)$ is the cross-entropy loss function. Based on the quantification study of feature transferability in deep convolutional networks (Yosinski et al., 2014), convolutional layers can learn generic features that are transferable across domains (Yosinski et al., 2014). Thus we opt to fine-tune the features of convolutional layers when transferring pre-trained deep models from source domain to target domain.

However, the literature findings also reveal that the deep features can reduce, but not remove, the cross-domain distribution discrepancy (Yosinski et al., 2014; Long et al., 2015;



(a) Joint Adaptation Network (JAN) (b) Adversarial Joint Adaptation Network (JAN-A)

Figure 1. The architectures of Joint Adaptation Network (JAN) (a) and its adversarial version (JAN-A) (b). Since deep features eventually transition from general to specific along the network, activations in multiple domain-specific layers $\mathcal{L}$ are not safely transferable. And the joint distributions of the activations $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$ in these layers should be adapted by JMMD minimization.

2016). The deep features in standard CNNs must eventually transition from general to specific along the network, and the transferability of features and classifiers decreases when the cross-domain discrepancy increases (Yosinski et al., 2014). In other words, even feed-forwarding the source and target domain data through the deep network for multilayer feature abstraction, the shifts in the joint distributions $P(\mathbf{X}^s, \mathbf{Y}^s)$ and $Q(\mathbf{X}^t, \mathbf{Y}^t)$ still linger in the activations $\mathbf{Z}^1, \ldots, \mathbf{Z}^{|\mathcal{L}|}$ of the higher network layers $\mathcal{L}$. Taking AlexNet (Krizhevsky et al., 2012) as an example, the activations in the higher fully-connected layers $\mathcal{L} = \{fc6, fc7, fc8\}$ are not safely transferable for domain adaptation (Yosinski et al., 2014). Note that the shift in the feature distributions $P(\mathbf{X}^s)$ and $Q(\mathbf{X}^t)$ mainly lingers in the feature layers $fc6, fc7$ while the shift in the label distributions $P(\mathbf{Y}^s)$ and $Q(\mathbf{Y}^t)$ mainly lingers in the classifier layer $fc8$. Thus we can use the joint distributions of the activations in layers $\mathcal{L}$, i.e. $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$ as good surrogates of the original joint distributions $P(\mathbf{X}^s, \mathbf{Y}^s)$ and $Q(\mathbf{X}^t, \mathbf{Y}^t)$, respectively. To enable unsupervised domain adaptation, we should find a way to match $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$.

4.1. Joint Maximum Mean Discrepancy

Many existing methods address transfer learning by bounding the target error with the source error plus a discrepancy between the marginal distributions $P(\mathbf{X}^s)$ and $Q(\mathbf{X}^t)$ of the source and target domains (Ben-David et al., 2010). The Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), as a kernel two-sample test statistic, has been widely applied to measure the discrepancy in marginal distributions $P(\mathbf{X}^s)$ and $Q(\mathbf{X}^t)$ (Tzeng et al., 2014; Long et al., 2015; 2016). To date MMD has not been used to measure the discrepancy in joint distributions $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$, possibly because MMD has not been directly defined for joint distributions by (Gretton et al., 2012) while in conventional shallow domain adaptation methods the joint distributions are not easy to manipulate and match.

Following the virtue of MMD (5), we use the Hilbert space embeddings of joint distributions (3) to measure the discrepancy of two joint distributions $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$. The resulting measure is called Joint Maximum Mean Discrepancy (JMMD), which is defined as

$$D_{\mathcal{L}}(P, Q) \triangleq \left\| \mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}(P) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}(Q) \right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|} \mathcal{H}^\ell}. \quad (8)$$

Based on the virtue of the kernel two-sample test theory (Gretton et al., 2012), we will have $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|}) = Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$ if and only if $D_{\mathcal{L}}(P, Q) = 0$. Given source domain $\mathcal{D}_s$ of n_s labeled points and target domain $\mathcal{D}_t$ of n_t unlabeled points drawn i.i.d. from P and Q respectively, the deep networks will generate activations in layers $\mathcal{L}$ as $\{(\mathbf{z}_i^{s1}, \ldots, \mathbf{z}_i^{s|\mathcal{L}|})\}_{i=1}^{n_s}$ and $\{(\mathbf{z}_j^{t1}, \ldots, \mathbf{z}_j^{t|\mathcal{L}|})\}_{j=1}^{n_t}$. The empirical estimate of $D_{\mathcal{L}}(P, Q)$ is computed as the squared distance between the empirical kernel mean embeddings as

$$\begin{aligned} \widehat{D}_{\mathcal{L}}(P, Q) = & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{s\ell} \right) \\ & + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{t\ell}, \mathbf{z}_j^{t\ell} \right) \\ & - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{t\ell} \right). \end{aligned} \quad (9)$$

Remark: Taking a close look on the objectives of MMD (6) and JMMD (9), we can find some interesting connections. The difference is that, for the activations $\mathbf{Z}^\ell$ in each layer $\ell \in \mathcal{L}$, instead of putting uniform weights on the kernel function $k^\ell(\mathbf{z}_i^\ell, \mathbf{z}_j^\ell)$ as in MMD, JMMD applies non-uniform weights, reflecting the influence of other variables in other layers $\mathcal{L} \backslash \ell$. This captures the full interactions between different variables in the joint distributions $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$, which is crucial for domain adaptation. All previous deep transfer learning methods (Tzeng et al., 2014; Long et al., 2015; Ganin & Lempitsky, 2015; Tzeng et al., 2015; Long et al., 2016) have not addressed this issue.

4.2. Joint Adaptation Networks

Denote by $\mathcal{L}$ the domain-specific layers where the activations are not safely transferable. We will formally reduce the discrepancy in the joint distributions of the activations

in layers $\mathcal{L}$, i.e. $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$. Note that the features in the lower layers of the network are transferable and hence will not require a further distribution matching. By integrating the JMMD (9) over the domain-specific layers $\mathcal{L}$ into the CNN error (7), the joint distributions are matched end-to-end with network training,

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q\right), \tag{10}$$

where $\lambda > 0$ is a tradeoff parameter of the JMMD penalty. As shown in Figure 1(a), we set $\mathcal{L} = \{fc6, fc7, fc8\}$ for the JAN model based on AlexNet (last three layers) while we set $\mathcal{L} = \{pool5, fc\}$ for the JAN model based on ResNet (last two layers), as these layers are tailored to task-specific structures, which are not safely transferable and should be jointly adapted by minimizing CNN error and JMMD (9).

A limitation of JMMD (9) is its quadratic complexity, which is inefficient for scalable deep transfer learning. Motivated by the unbiased estimate of MMD (Gretton et al., 2012), we derive a similar linear-time estimate of JMMD as follows,

$$\begin{aligned}\widehat{D}_{\mathcal{L}}\left(P, Q\right) = & \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}) \right) \\ & - \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{t\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{s\ell}) \right),\end{aligned} \tag{11}$$

where $n = n_s$. This linear-time estimate well fits the mini-batch stochastic gradient descent (SGD) algorithm. In each mini-batch, we sample the same number of source points and target points to eliminate the bias caused by domain size. This enables our models to scale linearly to large samples.

4.3. Adversarial Training for Optimal MMD

The MMD defined using the RKHS (6) has the advantage of not requiring a separate network to approximately maximize the original definition of MMD (5). But the original MMD (5) reveals that, in order to maximize the test power such that any two distributions can be distinguishable, we require the class of functions $f \in \mathcal{H}$ to be rich enough. Although (Gretton et al., 2012) shows that an universal RKHS is rich enough, such kernel-based MMD may suffer from vanishing gradients for low-bandwidth kernels. Moreover, it may be possible that some widely-used kernels are unable to capture very complex distances in high dimensional spaces such as natural images (Reddi et al., 2015; Arjovsky et al., 2017).

To circumvent the issues of vanishing gradients and non-rich function class of kernel-based MMD (6), we are enlightened by the original MMD (5) which fits the adversarial training in GANs (Goodfellow et al., 2014). We add multiple fully-connected layers parametrized by θ to the proposed JMMD (9) to make the function class of JMMD richer using neural network as shown in Figure 1(b). We maximize JMMD with respect to these new parameters θ to approach the virtue of the original MMD (5), that is, maximizing the test power of JMMD such that distributions of source and target domains are made more distinguishable (Sriperumbudur et al., 2009). This leads to a new adversarial joint adaptation network as

$$\min_f \max_\theta \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q; \theta\right). \tag{12}$$

Learning deep features by minimizing this more powerful JMMD, intuitively any shift in the joint distributions will be more easily identified by JMMD and then adapted by CNN.

Remark: This version of JAN shares the idea of domain-adversarial training with (Ganin & Lempitsky, 2015), but differs in that we use the JMMD as the domain adversary while (Ganin & Lempitsky, 2015) uses logistic regression. As pointed out in a very recent study (Arjovsky et al., 2017), our JMMD-adversarial network can be trained more easily.

5. Experiments

We evaluate the joint adaptation networks with state of the art transfer learning and deep learning methods. Codes and datasets are available at `http://github.com/thuml`.

5.1. Setup

Office-31 (Saenko et al., 2010) is a standard benchmark for domain adaptation in computer vision, comprising 4,652 images and 31 categories collected from three distinct domains: *Amazon* (**A**), which contains images downloaded from `amazon.com`, *Webcam* (**W**) and *DSLR* (**D**), which contain images respectively taken by web camera and digital SLR camera under different settings. We evaluate all methods across three transfer tasks **A** $\rightarrow$ **W**, **D** $\rightarrow$ **W** and **W** $\rightarrow$ **D**, which are widely adopted by previous deep transfer learning methods (Tzeng et al., 2014; Ganin & Lempitsky, 2015), and another three transfer tasks **A** $\rightarrow$ **D**, **D** $\rightarrow$ **A** and **W** $\rightarrow$ **A** as in (Long et al., 2015; 2016; Tzeng et al., 2015).

ImageCLEF-DA[1] is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge, which is organized by selecting the 12 common categories shared by the following three public datasets, each is considered as a domain: *Caltech-256* (**C**), *ImageNet ILSVRC 2012* (**I**), and *Pascal VOC 2012* (**P**). There are 50 images in each category and 600 images in each domain. We use all domain combinations and build 6 transfer tasks: **I** $\rightarrow$ **P**, **P** $\rightarrow$ **I**, **I** $\rightarrow$ **C**, **C** $\rightarrow$ **I**, **C** $\rightarrow$ **P**, and **P** $\rightarrow$ **C**. Different from *Office-31* where different domains are of different sizes, the three domains in ImageCLEF-DA are of equal size, which makes it a good complement to *Office-31* for more controlled experiments.

[1] `http://imageclef.org/2014/adaptation`

We compare with conventional and state of the art transfer learning and deep learning methods: Transfer Component Analysis (**TCA**) (Pan et al., 2011), Geodesic Flow Kernel (**GFK**) (Gong et al., 2012), Convolutional Neural Networks **AlexNet** (Krizhevsky et al., 2012) and **ResNet** (He et al., 2016), Deep Domain Confusion (**DDC**) (Tzeng et al., 2014), Deep Adaptation Network (**DAN**) (Long et al., 2015), Reverse Gradient (**RevGrad**) (Ganin & Lempitsky, 2015), and Residual Transfer Network (**RTN**) (Long et al., 2016). TCA is a transfer learning method based on MMD-regularized Kernel PCA. GFK is a manifold learning method that interpolates across an infinite number of intermediate subspaces to bridge domains. DDC is the first method that maximizes domain invariance by regularizing the adaptation layer of AlexNet using linear-kernel MMD (Gretton et al., 2012). DAN learns transferable features by embedding deep features of multiple task-specific layers to reproducing kernel Hilbert spaces (RKHSs) and matching different distributions optimally using multi-kernel MMD. RevGrad improves domain adaptation by making the source and target domains indistinguishable for a domain discriminator by adversarial training. RTN jointly learns transferable features and adaptive classifiers by deep residual learning (He et al., 2016).

We examine the influence of deep representations for domain adaptation by employing the breakthrough **AlexNet** (Krizhevsky et al., 2012) and the state of the art **ResNet** (He et al., 2016) for learning transferable deep representations. For AlexNet, we follow DeCAF (Donahue et al., 2014) and use the activations of layer $fc7$ as image representation. For ResNet (50 layers), we use the activations of the last feature layer $pool5$ as image representation. We follow standard evaluation protocols for unsupervised domain adaptation (Long et al., 2015; Ganin & Lempitsky, 2015). For both *Office-31* and *ImageCLEF-DA* datasets, we use all labeled source examples and all unlabeled target examples. We compare the average classification accuracy of each method on three random experiments, and report the standard error of the classification accuracies by different experiments of the same transfer task. We perform model selection by tuning hyper-parameters using transfer cross-validation (Zhong et al., 2010). For MMD-based methods and JAN, we adopt Gaussian kernel with bandwidth set to median pairwise squared distances on the training data (Gretton et al., 2012).

We implement all deep methods based on the **Caffe** framework, and fine-tune from Caffe-provided models of AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016), both are pre-trained on the ImageNet 2012 dataset. We fine-tune all convolutional and pooling layers and train the classifier layer via back propagation. Since the classifier is trained from scratch, we set its learning rate to be 10 times that of the other layers. We use mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate annealing strategy in RevGrad (Ganin & Lempitsky, 2015): the learning rate is not selected by a grid search due to high computational cost—it is adjusted during SGD using the following formula: $\eta_p = \frac{\eta_0}{(1+\alpha p)^\beta}$, where p is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01, \alpha = 10$ and $\beta = 0.75$, which is optimized to promote convergence and low error on the source domain. To suppress noisy activations at the early stages of training, instead of fixing the adaptation factor λ, we gradually change it from 0 to 1 by a progressive schedule: $\lambda_p = \frac{2}{1+\exp(-\gamma p)} - 1$, and $\gamma = 10$ is fixed throughout experiments (Ganin & Lempitsky, 2015). This progressive strategy significantly stabilizes parameter sensitivity and eases model selection for JAN and JAN-A.

5.2. Results

The classification accuracy results on the *Office-31* dataset for unsupervised domain adaptation based on AlexNet and ResNet are shown in Table 1. As fair comparison with identical evaluation setting, the results of DAN (Long et al., 2015), RevGrad (Ganin & Lempitsky, 2015), and RTN (Long et al., 2016) are directly reported from their published papers. The proposed JAN models outperform all comparison methods on most transfer tasks. It is noteworthy that JANs promote the classification accuracies substantially on hard transfer tasks, e.g. $\mathbf{D} \rightarrow \mathbf{A}$ and $\mathbf{W} \rightarrow \mathbf{A}$, where the source and target domains are substantially different and the source domain is smaller than the target domain, and produce comparable classification accuracies on easy transfer tasks, $\mathbf{D} \rightarrow \mathbf{W}$ and $\mathbf{W} \rightarrow \mathbf{D}$, where the source and target domains are similar (Saenko et al., 2010). The encouraging results highlight the key importance of joint distribution adaptation in deep neural networks, and suggest that JANs are able to learn more transferable representations for effective domain adaptation.

The results reveal several interesting observations. **(1)** Standard deep learning methods either outperform (AlexNet) or underperform (ResNet) traditional shallow transfer learning methods (TCA and GFK) using deep features (AlexNet-fc7 and ResNet-pool5) as input. And traditional shallow transfer learning methods perform better with more transferable deep features extracted by ResNet. This confirms the current practice that deep networks learn abstract feature representations, which can only reduce, but not remove, the domain discrepancy (Yosinski et al., 2014). **(2)** Deep transfer learning methods substantially outperform both standard deep learning methods and traditional shallow transfer learning methods. This validates that reducing the domain discrepancy by embedding domain-adaptation modules into deep networks (DDC, DAN, RevGrad, and RTN) can learn more transferable features. **(3)** The JAN models outperform previous methods by large margins and set new state of the art record. Different from all previous deep transfer learning methods that only adapt the marginal distributions based on independent feature layers (one layer for RevGrad and multilayer for DAN and RTN), JAN adapts the joint distribu-

Table 1. Classification accuracy (%) on *Office-31* dataset for unsupervised domain adaptation (AlexNet and ResNet)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky et al., 2012)	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA (Pan et al., 2011)	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK (Gong et al., 2012)	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DDC (Tzeng et al., 2014)	61.8±0.4	95.0±0.5	98.5±0.4	64.4±0.3	52.1±0.6	52.2±0.4	70.6
DAN (Long et al., 2015)	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN (Long et al., 2016)	73.3±0.3	**96.8**±0.2	**99.6**±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad (Ganin & Lempitsky, 2015)	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
JAN (ours)	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
JAN-A (ours)	**75.2**±0.4	96.6±0.2	**99.6**±0.1	**72.8**±0.3	57.5±0.2	**56.3**±0.2	**76.3**
ResNet (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA (Pan et al., 2011)	72.7±0.0	96.7±0.0	99.6±0.0	74.1±0.0	61.7±0.0	60.9±0.0	77.6
GFK (Gong et al., 2012)	72.8±0.0	95.0±0.0	98.2±0.0	74.5±0.0	63.4±0.0	61.0±0.0	77.5
DDC (Tzeng et al., 2014)	75.6±0.2	96.0±0.2	98.2±0.1	76.5±0.3	62.2±0.4	61.5±0.5	78.3
DAN (Long et al., 2015)	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN (Long et al., 2016)	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
RevGrad (Ganin & Lempitsky, 2015)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN (ours)	85.4±0.3	**97.4**±0.2	**99.8**±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
JAN-A (ours)	**86.0**±0.4	96.7±0.3	99.7±0.1	**85.1**±0.4	**69.2**±0.4	**70.7**±0.5	**84.6**

Table 2. Classification accuracy (%) on *ImageCLEF-DA* for unsupervised domain adaptation (AlexNet and ResNet)

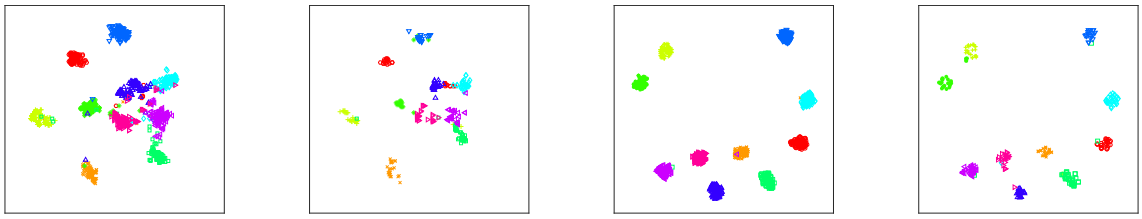
Method	I → P	P → I	I → C	C → I	C → P	P → C	Avg
AlexNet (Krizhevsky et al., 2012)	66.2±0.2	70.0±0.2	84.3±0.2	71.3±0.4	59.3±0.5	84.5±0.3	73.9
DAN (Long et al., 2015)	67.3±0.2	80.5±0.3	87.7±0.3	76.0±0.3	61.6±0.3	88.4±0.2	76.9
RTN (Long et al., 2016)	**67.4**±0.3	81.3±0.3	89.5±0.4	78.0±0.2	62.0±0.2	89.1±0.1	77.9
JAN (ours)	67.2±0.5	**82.8**±0.4	**91.3**±0.5	**80.0**±0.5	**63.5**±0.4	**91.0**±0.4	**79.3**
ResNet (He et al., 2016)	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN (Long et al., 2015)	74.5±0.4	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
RTN (Long et al., 2016)	74.6±0.3	85.8±0.1	94.3±0.1	85.9±0.3	71.7±0.3	91.2±0.4	83.9
JAN (ours)	**76.8**±0.4	**88.0**±0.2	**94.7**±0.2	**89.5**±0.3	**74.2**±0.3	**91.7**±0.3	**85.8**

tions of network activations in all domain-specific layers to fully correct the shifts in joint distributions across domains. Although both JAN and DAN (Long et al., 2015) adapt multiple domain-specific layers, the improvement from DAN to JAN is crucial for the domain adaptation performance: JAN uses a JMMD penalty to reduce the shift in the joint distributions of multiple task-specific layers, which reflects the shift in the joint distributions of input features and output labels; DAN needs multiple MMD penalties, each independently reducing the shift in the marginal distribution of each layer, assuming feature layers and classifier layer are independent.

By going from AlexNet to extremely deep ResNet, we can attain a more in-depth understanding of feature transferability. **(1)** ResNet-based methods outperform AlexNet-based methods by large margins. This validates that very deep convolutional networks, e.g. VGGnet (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), and ResNet, not only learn better representations for general vision tasks but also learn more transferable representations for domain adaptation. **(2)** The JAN models significantly outperform ResNet-based methods, revealing that even very deep networks can only reduce, but not remove, the domain discrepancy. **(3)** The boost of JAN over ResNet is more significant than the improvement of JAN over AlexNet. This implies that JAN can benefit from more transferable representations.

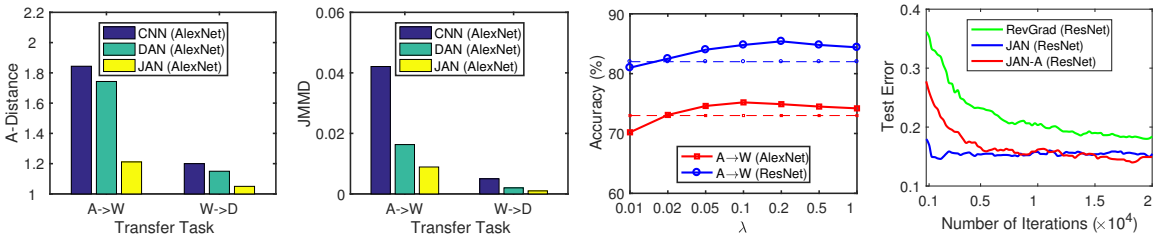
The great aspect of JAN is that via the kernel trick there is no need to train a separate network to maximize the MMD criterion (5) for the ball of a RKHS. However, this has the disadvantage that some kernels used in practice are unsuitable for capturing very complex distances in high dimensional spaces such as natural images (Arjovsky et al., 2017). The JAN-A model significantly outperforms the previous domain adversarial deep network (Ganin & Lempitsky, 2015). The improvement from JAN to JAN-A also demonstrates the benefit of adversarial training for optimizing the JMMD in a richer function class. By maximizing the JMMD criterion with respect to a separate network, JAN-A can maximize the distinguishability of source and target distributions. Adapting domains against deep features where their distributions maximally differ, we can enhance the feature transferability.

The three domains in *ImageCLEF-DA* are more balanced than those of *Office-31*. With these more balanced transfer tasks, we are expecting to testify whether transfer learning improves when domain sizes do not change. The classification accuracy results based on both AlexNet and ResNet are shown in Table 2. The JAN models outperform comparison methods on most transfer tasks, but by less improvements. This means the difference in domain sizes may cause shift.



(a) DAN: *Source*=**A** (b) DAN: *Target*=**W** (c) JAN: *Source*=**A** (d) JAN: *Target*=**W**

Figure 2. The t-SNE visualization of network activations (ResNet) generated by DAN (a)(b) and JAN (c)(d), respectively.



(a) $\mathcal{A}$-distance (b) JMMD (c) Accuracy w.r.t. λ (d) Convergence

Figure 3. Analysis: (a) $\mathcal{A}$-distance; (b) JMMD; (c) parameter sensitivity of λ; (d) convergence (dashed lines show best baseline results).

5.3. Analysis

Feature Visualization: We visualize in Figures 2(a)–2(d) the network activations of task $\mathbf{A} \rightarrow \mathbf{W}$ learned by DAN and JAN respectively using t-SNE embeddings (Donahue et al., 2014). Compared with the activations given by DAN in Figure 2(a)–2(b), the activations given by JAN in Figures 2(c)–2(d) show that the target categories are discriminated much more clearly by the JAN source classifier. This suggests that the adaptation of joint distributions of multilayer activations is a powerful approach to unsupervised domain adaptation.

Distribution Discrepancy: The theory of domain adaptation (Ben-David et al., 2010; Mansour et al., 2009) suggests $\mathcal{A}$-distance as a measure of distribution discrepancy, which, together with the source risk, will bound the target risk. The proxy $\mathcal{A}$-distance is defined as $d_{\mathcal{A}} = 2\,(1 - 2\epsilon)$, where ϵ is the generalization error of a classifier (e.g. kernel SVM) trained on the binary problem of discriminating the source and target. Figure 3(a) shows $d_{\mathcal{A}}$ on tasks $\mathbf{A} \rightarrow \mathbf{W}$, $\mathbf{W} \rightarrow \mathbf{D}$ with features of CNN, DAN, and JAN. We observe that $d_{\mathcal{A}}$ using JAN features is much smaller than $d_{\mathcal{A}}$ using CNN and DAN features, which suggests that JAN features can close the cross-domain gap more effectively. As domains **W** and **D** are very similar, $d_{\mathcal{A}}$ of task $\mathbf{W} \rightarrow \mathbf{D}$ is much smaller than that of $\mathbf{A} \rightarrow \mathbf{W}$, which explains better accuracy of $\mathbf{W} \rightarrow \mathbf{D}$.

A limitation of the $\mathcal{A}$-distance is that it cannot measure the cross-domain discrepancy of joint distributions, which is addressed by the proposed JMMD (9). We compute JMMD (9) across domains using CNN, DAN and JAN activations respectively, based on the features in $fc7$ and ground-truth labels in $fc8$ (the target labels are not used for model training). Figure 3(b) shows that JMMD using JAN activations is much smaller than JMMD using CNN and DAN activations, which validates that JANs successfully reduce the shifts in joint distributions to learn more transferable representations.

Parameter Sensitivity: We check the sensitivity of JMMD parameter λ, i.e. the maximum value of the relative weight for JMMD. Figure 3(c) demonstrates the transfer accuracy of JAN based on AlexNet and ResNet respectively, by varying $\lambda \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$ on task $\mathbf{A} \rightarrow \mathbf{W}$. The accuracy of JAN first increases and then decreases as λ varies and shows a bell-shaped curve. This confirms the motivation of deep learning and joint distribution adaptation, as a proper trade-off between them enhance transferability.

Convergence Performance: As JAN and JAN-A involve adversarial training procedures, we testify their convergence performance. Figure 3(d) demonstrates the test errors of different methods on task $\mathbf{A} \rightarrow \mathbf{W}$, which suggests that JAN converges fastest due to nonparametric JMMD while JAN-A has similar convergence speed as RevGrad with significantly improved accuracy in the whole procedure of convergence.

6. Conclusion

This paper presented a novel approach to deep transfer learning, which enables end-to-end learning of transferable representations. Unlike previous methods that match the marginal distributions of features across domains, the proposed approach reduces the shift in joint distributions of the network activations of multiple task-specific layers, which approximates the shift in the joint distributions of input features and output labels. The discrepancy between joint distributions can be computed by embedding the joint distributions in a tensor-product Hilbert space, which can be scaled linearly to large samples and be implemented in most deep networks. Experiments testified the efficacy of the proposed approach.

Acknowledgments

We thank Zhangjie Cao for conducting part of experiments. This work was supported by NSFC (61502265, 61325008), National Key R&D Program of China (2016YFB1000701, 2015BAF32B01), and Tsinghua TNList Lab Key Projects.

References

Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

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

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

Bousmalis, Konstantinos, Trigeorgis, George, Silberman, Nathan, Krishnan, Dilip, and Erhan, Dumitru. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 343–351, 2016.

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

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

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

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International Conference on Machine Learning (ICML)*, 2011.

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

Gong, B., Grauman, K., and Sha, F. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2013.

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

Gopalan, R., Li, R., and Chellappa, R. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13:723–773, 2012.

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

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

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Schölkopf, B. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

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

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

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

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

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

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

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. The MIT Press, 2009.

Reddi, Sashank J, Ramdas, Aaditya, Póczos, Barnabás, Singh, Aarti, and Wasserman, Larry A. On the high dimensional power of a linear-time two sample test under mean-shift alternatives. In *Artificial Intelligence and Statistics Conference (AISTATS)*, 2015.

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

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

Smola, Alex, Gretton, Arthur, Song, Le, and Schölkopf, Bernhard. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory (ALT)*, pp. 13–31. Springer, 2007.

Song, L., Huang, J., Smola, A., and Fukumizu, K. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *International Conference on Machine Learning (ICML)*, 2009.

Song, Le and Dai, Bo. Robust low rank kernel embeddings of multivariate distributions. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3228–3236, 2013.

Song, Le, Boots, Byron, Siddiqi, Sajid M, Gordon, Geoffrey J, and Smola, Alex. Hilbert space embeddings of hidden markov models. In *International Conference on Machine Learning (ICML)*, 2010.

Song, Le, Fukumizu, Kenji, and Gretton, Arthur. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Lanckriet, G., and Schölkopf, B. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Sriperumbudur, Bharath K, Gretton, Arthur, Fukumizu, Kenji, Schölkopf, Bernhard, and Lanckriet, Gert RG. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research (JMLR)*, 11(Apr):1517–1561, 2010.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

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

Tzeng, Eric, Hoffman, Judy, Saenko, Kate, and Darrell, Trevor. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

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

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

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

Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pp. 547–562. Springer, 2010.