Learning Transferable Features with Deep Adaptation Networks

Mingsheng Long[†♯] MINGSHENG@TSINGHUA.EDU.CN
Yue Cao[†] YUE-CAO14@MAILS.TSINGHUA.EDU.CN
Jianmin Wang[†] JIMWANG@TSINGHUA.EDU.CN
Michael I. Jordan[♯] JORDAN@BERKELEY.EDU
[†]School of Software, TNList Lab for Info. Sci. & Tech., Institute for Data Science, Tsinghua University, China
[♯]Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA

Abstract

Recent studies reveal that a deep neural network can learn transferable features which generalize well to novel tasks for domain adaptation. However, as deep features eventually transition from general to specific along the network, the feature transferability drops significantly in higher layers with increasing domain discrepancy. Hence, it is important to formally reduce the dataset bias and enhance the transferability in task-specific layers. In this paper, we propose a new Deep Adaptation Network (DAN) architecture, which generalizes deep convolutional neural network to the domain adaptation scenario. In DAN, hidden representations of all task-specific layers are embedded in a reproducing kernel Hilbert space where the mean embeddings of different domain distributions can be explicitly matched. The domain discrepancy is further reduced using an optimal multi-kernel selection method for mean embedding matching. DAN can learn transferable features with statistical guarantees, and can scale linearly by unbiased estimate of kernel embedding. Extensive empirical evidence shows that the proposed architecture yields state-of-the-art image classification error rates on standard domain adaptation benchmarks.

1. Introduction

The generalization error of supervised learning machines with limited training samples will be unsatisfactorily large, while manual labeling of sufficient training data for diverse application domains may be prohibitive. Therefore, there is incentive to establishing effective algorithms to reduce the labeling cost, typically by leveraging off-the-shelf labeled data from relevant source domains to the target domains. Domain adaptation addresses the problem that we have data from two related domains but under different distributions. The domain discrepancy poses a major obstacle in adapting predictive models across domains. For example, an object recognition model trained on manually annotated images may not generalize well on testing images under substantial variations in the pose, occlusion, or illumination. Domain adaptation establishes knowledge transfer from the labeled source domain to the unlabeled target domain by exploring domain-invariant structures that bridge different domains of substantial distribution discrepancy (Pan & Yang, 2010).

One of the main approaches to establishing knowledge transfer is to learn domain-invariant models from data, which can bridge the source and target domains in an isomorphic latent feature space. In this direction, a fruitful line of prior work has focused on learning shallow features by jointly minimizing a distance metric of domain discrepancy (Pan et al., 2011; Long et al., 2013; Baktashmotlagh et al., 2013; Gong et al., 2013; Zhang et al., 2013; Ghifary et al., 2014; Wang & Schneider, 2014). However, recent studies have shown that deep neural networks can learn more transferable features for domain adaptation (Glorot et al., 2011; Donahue et al., 2014; Yosinski et al., 2014), which produce breakthrough results on some domain adaptation datasets. Deep neural networks are able to disentangle exploratory factors of variations underlying the data samples, and group features hierarchically in accordance with their relatedness to invariant factors, making representations robust to noise.

While deep neural networks are more powerful for learning general and transferable features, the latest findings also reveal that the deep features must eventually transition from general to specific along the network, and feature transferability drops significantly in higher layers with increasing domain discrepancy. In other words, the features computed in higher layers of the network must depend greatly on the specific dataset and task (Yosinski et al., 2014), which are task-specific features and are not safely transferable to

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

novel tasks. Another curious phenomenon is that disentangling the variational factors in higher layers of the network may enlarge the domain discrepancy, as different domains with the new deep representations become more "compact" and are more mutually distinguishable (Glorot et al., 2011). Although deep features are salient for discrimination, enlarged dataset bias may deteriorate domain adaptation performance, resulting in statistically *unbounded* risk for the target tasks (Mansour et al., 2009; Ben-David et al., 2010).

Inspired by the literature's latest understanding about the transferability of deep neural networks, we propose in this paper a new Deep Adaptation Network (DAN) architecture, which generalizes deep convolutional neural network to the domain adaptation scenario. The main idea of this work is to enhance the feature transferability in the task-specific layers of the deep neural network by explicitly reducing the domain discrepancy. To establish this goal, the hidden representations of all the task-specific layers are embedded to a reproducing kernel Hilbert space where the mean embeddings of different domain distributions can be explicitly matched. As mean embedding matching is sensitive to the kernel choices, an optimal multi-kernel selection procedure is devised to further reduce the domain discrepancy. In addition, we implement a linear-time unbiased estimate of the kernel mean embedding to enable scalable training, which is very desirable for deep learning. Finally, as deep models pre-trained with large-scale repositories such as ImageNet (Russakovsky et al., 2014) are representative for general-purpose tasks (Yosinski et al., 2014; Hoffman et al., 2014), the proposed DAN model is trained by fine-tuning from the AlexNet model (Krizhevsky et al., 2012) pre-trained on ImageNet, which is implemented in Caffe (Jia et al., 2014). Comprehensive empirical evidence demonstrates that the proposed architecture outperforms state-of-the-art results evaluated on the standard domain adaptation benchmarks.

The contributions of this paper are summarized as follows. (1) We propose a novel deep neural network architecture for domain adaptation, in which *all* the layers corresponding to task-specific features are adapted in a layerwise manner, hence benefiting from "deep adaptation." (2) We explore *multiple* kernels for adapting deep representations, which substantially enhances adaptation effectiveness compared to single kernel methods. Our model can yield unbiased deep features with statistical guarantees.

2. Related Work

A related literature is transfer learning (Pan & Yang, 2010), which builds models that bridge different domains or tasks, explicitly taking domain discrepancy into consideration. Transfer learning aims to mitigate the effort of manual labeling for machine learning (Pan et al., 2011; Gong et al., 2013; Zhang et al., 2013; Wang & Schneider, 2014) and computer vision (Saenko et al., 2010; Gong et al., 2012; Baktashmotlagh et al., 2013; Long et al., 2013), etc. It is widely recognized that the domain discrepancy in the probability distributions of different domains should be formally measured and reduced. The major bottleneck is how to match different domain distributions effectively. Most existing methods learn a new shallow representation model in which the domain discrepancy can be explicitly reduced. However, without learning deep features which can suppress domain-specific factors, the transferability of shallow features could be limited by the task-specific variability.

Deep neural networks learn nonlinear representations that disentangle and hide different explanatory factors of variation behind data samples (Bengio et al., 2013). The learned deep representations manifest invariant factors underlying different populations and are transferable from the original tasks to similar novel tasks (Yosinski et al., 2014). Hence, deep neural networks have been explored for domain adaptation (Glorot et al., 2011; Chen et al., 2012), multimodal and multi-source learning problems (Ngiam et al., 2011; Ge et al., 2013), where significant performance gains have been obtained. However, all these methods depend on the assumption that deep neural networks can learn invariant representations that are transferable across different tasks. In reality, the domain discrepancy can be alleviated, but not removed, by deep neural networks (Glorot et al., 2011). Dataset shift has posed a bottleneck to the transferability of deep networks, resulting in statistically *unbounded* risk for target tasks (Mansour et al., 2009; Ben-David et al., 2010).

Our work is primarily motivated by Yosinski et al. (2014), which comprehensively explores feature transferability of deep convolutional neural networks. The method focuses on a different scenario where the learning tasks are different across domains, hence it requires sufficient target labeled examples such that the source network can be fine-tuned to the target task. In many real problems, labeled data is usually limited especially for a novel target task, hence the method cannot be directly applicable to domain adaptation. There are several very recent efforts in learning domain-invariant features in the context of shallow neural networks (Ajakan et al., 2014; Ghifary et al., 2014). Due to the limited capacity of shallow architectures, the performance of these proposals does not surpass deep CNN (Krizhevsky et al., 2012). Tzeng et al. (2014) proposed a DDC model that adds an adaptation layer and a dataset shift loss to the deep CNN for learning a domain-invariant representation. While performance was improved, DDC only adapts a single layer of the network, which may be restrictive in that there are multiple layers where the hidden features are not transferable (Yosinski et al., 2014). DDC is also limited by suboptimal kernel matching of probability distributions (Gretton et al., 2012b) and its quadratic computational cost that restricts transferability and scalability.

3. Deep Adaptation Networks

In unsupervised domain adaptation, we are given a *source* domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled examples, and a *target* domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ with n_t unlabeled examples. The source domain and target domain are characterized by probability distributions p and q, respectively. We aim to construct a deep neural network which is able to learn transferable features that bridge the cross-domain discrepancy, and build a classifier $y = \theta(\mathbf{x})$ which can minimize target risk $\epsilon_t(\theta) = \Pr_{(\mathbf{x},y)\sim q}[\theta(\mathbf{x}) \neq y]$ using source supervision. In semi-supervised adaptation where the target has a small number of labeled examples, we denote by $\mathcal{D}_a = \{(\mathbf{x}_i^a, y_i^a)\}$ the n_a annotated examples of source and target domains.

3.1. Model

MK-MMD Domain adaptation is challenging in that the target domain has no (or only limited) labeled information. To approach this problem, many existing methods aim to bound the target error by the source error plus a discrepancy metric between the source and the target (Ben-David et al., 2010). Two classes of statistics have been explored for the *two-sample* testing, where acceptance or rejection decisions are made for a null hypothesis $p = q$, given samples generated respectively from p and q: *energy distances* and *maximum mean discrepancies* (MMD) (Sejdinovic et al., 2013). In this paper, we focus on the multiple kernel variant of MMD (MK-MMD) proposed by Gretton et al. (2012b), which is formalized to jointly maximize the two-sample test power and minimize the Type II error, i.e., the failure of rejecting a false null hypothesis.

Denote by $\mathcal{H}_k$ be the reproducing kernel Hilbert space (RKHS) endowed with a characteristic kernel k. The *mean embedding* of distribution p in $\mathcal{H}_k$ is a unique element $\mu_k(p)$ such that $\mathbf{E}_{\mathbf{x}\sim p} f(\mathbf{x}) = \langle f(\mathbf{x}), \mu_k(p)\rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$. The MK-MMD $d_k(p,q)$ between probability distributions p and q is defined as the RKHS distance between the mean embeddings of p and q. The squared formulation of MK-MMD is defined as

$$d_k^2(p,q) \triangleq \left\| \mathbf{E}_p[\phi(\mathbf{x}^s)] - \mathbf{E}_q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2. \quad (1)$$

The most important property is that $p = q$ iff $d_k^2(p,q) = 0$ (Gretton et al., 2012a). The characteristic kernel associated with the feature map ϕ, $k(\mathbf{x}^s, \mathbf{x}^t) = \langle \phi(\mathbf{x}^s), \phi(\mathbf{x}^t)\rangle$, is defined as the convex combination of m PSD kernels $\{k_u\}$,

$$\mathcal{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geqslant 0, \forall u \right\}, \quad (2)$$

where the constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that the derived multi-kernel k is characteristic. As studied theoretically in Gretton et al. (2012b), the kernel adopted for the mean embeddings of p and q is critical to ensure the test power and low test error. The multi-kernel k can leverage different kernels to enhance MK-MMD test, leading to a principled method for optimal kernel selection.

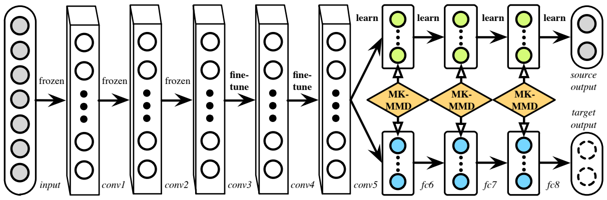


Figure 1. The DAN architecture for learning transferable features. Since deep features eventually transition from general to specific along the network, (1) the features extracted by convolutional layers $conv1$–$conv3$ are general, hence these layers are frozen, (2) the features extracted by layers $conv4$–$conv5$ are slightly less transferable, hence these layers are learned via fine-tuning, and (3) fully connected layers $fc6$–$fc8$ are tailored to fit specific tasks, hence they are not transferable and should be adapted with MK-MMD.

One of the feasible strategies for controlling the domain discrepancy is to find an abstract feature representation through which the source and target domains are similar (Ben-David et al., 2010). Although this idea has been explored in several papers (Pan et al., 2011; Zhang et al., 2013; Wang & Schneider, 2014), to date there has been no attempt to enhance the transferability of feature representation via MK-MMD in deep neural networks.

Deep Adaptation Networks (DAN) In this paper, we explore the idea of MK-MMD-based adaptation for learning transferable features in deep networks. We start with deep convolutional neural networks (CNN) (Krizhevsky et al., 2012), a strong model when it is adapted to novel tasks (Donahue et al., 2014; Hoffman et al., 2014). The main challenge is that the target domain has no or just limited labeled information, hence directly adapting CNN to the target domain via fine-tuning is impossible or is prone to over-fitting. With the idea of domain adaptation, we are targeting a deep adaptation network (DAN) that can exploit both source-labeled data and target-unlabeled data. Figure 1 gives an illustration of the proposed DAN model.

We extend the AlexNet architecture (Krizhevsky et al., 2012), which is comprised of five convolutional layers ($conv1$–$conv5$) and three fully connected layers ($fc6$–$fc8$). Each fc layer ℓ learns a nonlinear mapping $\mathbf{h}_i^\ell = f^\ell\left(\mathbf{W}^\ell \mathbf{h}_i^{\ell-1} + \mathbf{b}^\ell\right)$, where $\mathbf{h}_i^\ell$ is the ℓth layer hidden representation of point $\mathbf{x}_i$, $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ are the weights and bias of the ℓth layer, and f^ℓ is the activation, taking as rectifier units $f^\ell(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x})$ for hidden layers or softmax units $f^\ell(\mathbf{x}) = e^{\mathbf{x}} / \sum_{j=1}^{|\mathbf{x}|} e^{x_j}$ for the output layer. Letting

$\Theta = \left\{ \mathbf{W}^\ell, \mathbf{b}^\ell \right\}_{\ell=1}^{l}$ denote the set of all CNN parameters, the empirical risk of CNN is

$$\min_\Theta \frac{1}{n_a} \sum_{i=1}^{n_a} J\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right), \tag{3}$$

where J is the cross-entropy loss function, and $\theta\left(\mathbf{x}_i^a\right)$ is the conditional probability that the CNN assigns $\mathbf{x}_i^a$ to label y_i^a. We will not discuss how to compute the convolutional layers as we will not impose distribution-adaptation regularization in those layers, given that the convolutional layers can learn generic features that tend to be transferable in layers $conv1$–$conv3$ and are slightly domain-biased in $conv4$–$conv5$ (Yosinski et al., 2014). Hence, when adapting the pre-trained AlexNet to the target, we opt to freeze $conv1$–$conv3$ and fine-tune $conv4$–$conv5$ to preserve the efficacy of fragile co-adaptation (Hinton et al., 2012).

In standard CNNs, deep features must eventually transition from general to specific by the last layer of the network, and the transferability gap grows with the domain discrepancy and becomes particularly large when transferring the higher layers $fc6$–$fc8$ (Yosinski et al., 2014). In other words, the fc layers are tailored to their original task at the expense of degraded performance on the target task, hence they cannot be directly transferred to the target domain via fine-tuning with limited target supervision. In this paper, we fine-tune CNN on the source labeled examples and require the distributions of the source and target to become similar under the hidden representations of fully connected layers $fc6$–$fc8$. This can be realized by adding an MK-MMD-based multi-layer adaptation regularizer (1) to the CNN risk (3):

$$\min_\Theta \frac{1}{n_a} \sum_{i=1}^{n_a} J\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right), \tag{4}$$

where $\lambda > 0$ is a penalty parameter, l_1 and l_2 are layer indices between which the regularizer is effective. In our implementation of DAN, we set $l_1 = 6$ and $l_2 = 8$, although different configurations are also possible, depending on the size of the labeled source dataset and the number of parameters in the layers that are to be fine-tuned. $\mathcal{D}_*^\ell = \left\{\mathbf{h}_i^{*\ell}\right\}$ is the ℓth layer hidden representation for the source and target examples, and $d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right)$ is the MK-MMD between the source and target evaluated on the ℓth layer representation.

Training a deep CNN requires a large amount of labeled data, which is prohibitive for many domain adaptation problems, hence we start with an AlexNet model pre-trained on ImageNet 2012 and fine-tune it as in Yosinski et al. (2014). With the proposed DAN optimization framework (4), we are able to learn transferable features from a source domain to a related target domain. The learned representation can both be salient benefiting from CNN, and unbiased thanks to MK-MMD. Two important advantages that distinguish DAN from relevant literature are: (1) *multi-layer* adaptation. As revealed by (Yosinski et al., 2014), feature transferability gets worse on $conv4$–$conv5$ and significantly drops on $fc6$–$fc8$, hence it is critical to adapt multiple layers instead of only one layer. In other words, adapting a single layer cannot undo the dataset bias between the source and the target, since there are other layers that are not transferable. Another benefit of multi-layer adaptation is that by jointly adapting the representation layers and the classifier layer, we could essentially bridge the domain discrepancy underlying *both* the marginal distribution and the conditional distribution, which is crucial for domain adaptation (Zhang et al., 2013). (2) *multi-kernel* adaptation. As pointed out by Gretton et al. (2012b), kernel choice is critical to the testing power of MMD since different kernels may embed probability distributions in different RKHSs where different orders of sufficient statistics can be emphasized. This is crucial for moment matching, which is not well explored by previous domain adaptation methods.

3.2. Algorithm

Learning Θ Using the kernel trick, MK-MMD (1) can be computed as the expectation of kernel functions $d_k^2\left(p, q\right) = \mathbf{E}_{\mathbf{x}^s\mathbf{x}'^s} k(\mathbf{x}^s, \mathbf{x}'^s) + \mathbf{E}_{\mathbf{x}^t\mathbf{x}'^t} k(\mathbf{x}^t, \mathbf{x}'^t) - 2\mathbf{E}_{\mathbf{x}^s\mathbf{x}^t} k(\mathbf{x}^s, \mathbf{x}^t)$, where $\mathbf{x}^s, \mathbf{x}'^s \overset{iid}{\sim} p$, $\mathbf{x}^t, \mathbf{x}'^t \overset{iid}{\sim} q$, and $k \in \mathcal{K}$. However, this computation incurs a complexity of $O(n^2)$, which is rather undesirable for deep CNNs, as the power of deep neural networks largely derives from learning with large-scale datasets. Moreover, the summation over pairwise similarities between data points makes mini-batch stochastic gradient descent (SGD) more difficult, whereas mini-batch SGD is crucial to the training effectiveness of deep networks. While prior work based on MMD (Pan et al., 2011; Tzeng et al., 2014) rarely addresses this issue, we believe it is critical in the context of deep learning. In this paper, we adopt the unbiased estimate of MK-MMD (Gretton et al., 2012b) which can be computed with linear complexity. More specifically, $d_k^2\left(p, q\right) = \frac{2}{n_s} \sum_{i=1}^{n_s/2} g_k\left(\mathbf{z}_i\right)$, where we denote quad-tuple $\mathbf{z}_i \triangleq \left(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t\right)$, and evaluate multi-kernel function k on each quad-tuple $\mathbf{z}_i$ by $g_k\left(\mathbf{z}_i\right) \triangleq k(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s) + k(\mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t)$. This approach computes an expectation of independent variables as in (1) with cost $O(n)$.

When we train deep CNN by mini-batch SGD, we only need to consider the gradient of objective (4) with respect to each data point $\mathbf{x}_i$. Since the linear-time MK-MMD takes a nice summation form that can be readily decoupled into the sum of $g_k(\mathbf{z}_i)$'s, we only need to compute the gradients $\frac{\partial g_k(\mathbf{z}_i^\ell)}{\partial \Theta^\ell}$ for the quad-tuple $\mathbf{z}_i^\ell = \left(\mathbf{h}_{2i-1}^{s\ell}, \mathbf{h}_{2i}^{s\ell}, \mathbf{h}_{2i-1}^{t\ell}, \mathbf{h}_{2i}^{t\ell}\right)$ of the ℓth layer hidden representation. To be consistent with the gradient of MK-MMD, we need to compute the corresponding gradient of CNN risk $\frac{\partial J(\mathbf{z}_i)}{\partial \Theta^\ell}$, where $J\left(\mathbf{z}_i\right) =$

$\sum_{i'} J\left(\theta\left(\mathbf{x}_{i'}^a\right), y_{i'}^a\right)$, and $\{(\mathbf{x}_{i'}^a, y_{i'}^a)\}$ indicates the labeled examples in quad-tuple $\mathbf{z}_i$—for instance, in unsupervised adaptation where the target domain has no labeled data, we have $\{(\mathbf{x}_{i'}^a, y_{i'}^a)\} = \{(\mathbf{x}_{2i-1}^s, y_{2i-1}^s), (\mathbf{x}_{2i}^s, y_{2i}^s)\}$. To perform a mini-batch update, we compute the gradient of objective (4) with respect to the ℓth layer parameter Θ^ℓ as

$$\nabla_{\Theta^\ell} = \frac{\partial J\left(\mathbf{z}_i\right)}{\partial \Theta^\ell} + \lambda \frac{\partial g_k\left(\mathbf{z}_i^\ell\right)}{\partial \Theta^\ell}. \tag{5}$$

Such a mini-batch SGD can be easily implemented within the Caffe framework for CNNs (Jia et al., 2014). Given kernel k as the linear combination of m Gaussian kernels $\{k_u\left(\mathbf{x}_i, \mathbf{x}_j\right) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\gamma_u}\}$, the gradient $\frac{\partial g_k\left(\mathbf{z}_i^\ell\right)}{\partial \Theta^\ell}$ can be readily computed using the chain rule. For instance,

$$\begin{aligned}\frac{\partial k(\mathbf{h}_{2i-1}^{s\ell}, \mathbf{h}_{2i}^{t\ell})}{\partial \mathbf{W}^\ell} = & -\sum_{u=1}^{m} \frac{2\beta_u}{\gamma_u} k_u\left(\mathbf{h}_{2i-1}^{s\ell}, \mathbf{h}_{2i}^{t\ell}\right) \\ & \times \left(\mathbf{h}_{2i-1}^{s\ell} - \mathbf{h}_{2i}^{t\ell}\right) \\ & \times \left(\mathbb{I}\left[\mathbf{h}_{2i-1}^{s(\ell-1)}\right] - \mathbb{I}\left[\mathbf{h}_{2i}^{t(\ell-1)}\right]\right)^{\mathsf{T}},\end{aligned} \tag{6}$$

where the last row computes the gradient of the ℓth layer rectifier units, with $\mathbb{I}$ being defined as an indicator such that $\mathbb{I}\left[\mathbf{h}_{ji}^{\ell-1}\right] = \mathbf{h}_{ji}^{\ell-1}$ if $\mathbf{W}_{j\cdot}^\ell \mathbf{h}_i^{\ell-1} + \mathbf{b}_j^\ell \geqslant 0$, else $\mathbb{I}\left[\mathbf{h}_{ji}^{\ell-1}\right] = 0$.

Learning $\boldsymbol{\beta}$ The proposed multi-layer adaptation regularizer performs layerwise matching by MK-MMD, hence we seek to learn optimal kernel parameter $\boldsymbol{\beta}$ for MK-MMD by jointly maximizing the test power and minimizing the Type II error (Gretton et al., 2012b), leading to the optimization

$$\max_{k \in \mathcal{K}} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right) \sigma_k^{-2}, \tag{7}$$

where $\sigma_k^2 = \mathbf{E}_{\mathbf{z}} g_k^2\left(\mathbf{z}\right) - \left[\mathbf{E}_{\mathbf{z}} g_k\left(\mathbf{z}\right)\right]^2$ is estimation variance. Letting $\mathbf{d} = (d_1, d_2, \ldots, d_m)^{\mathsf{T}}$, each d_u is MMD via kernel k_u. Covariance $\mathbf{Q} = \operatorname{cov}\left(g_k\right) \in \mathbb{R}^{m \times m}$ can be computed in $O(m^2 n)$ cost, i.e. $\mathbf{Q}_{uu'} = \frac{4}{n_s} \sum_{i=1}^{n_s/4} g_{k_u}^{\Delta}\left(\bar{\mathbf{z}}_i\right) g_{k_{u'}}^{\Delta}\left(\bar{\mathbf{z}}_i\right)$, where $\bar{\mathbf{z}}_i \triangleq \left(\mathbf{z}_{2i-1}, \mathbf{z}_{2i}\right)$ and $g_{k_u}^{\Delta}\left(\bar{\mathbf{z}}_i\right) \triangleq g_{k_u}\left(\mathbf{z}_{2i-1}\right) - g_{k_u}\left(\mathbf{z}_{2i}\right)$. Hence (7) reduces to a quadratic program (QP),

$$\min_{\mathbf{d}^{\mathsf{T}}\boldsymbol{\beta}=1, \boldsymbol{\beta} \geqslant \mathbf{0}} \boldsymbol{\beta}^{\mathsf{T}}\left(\mathbf{Q} + \varepsilon \mathbf{I}\right)\boldsymbol{\beta}, \tag{8}$$

where $\varepsilon = 10^{-3}$ is a small regularizer to make the problem well-defined. By solving (8), we obtain a multi-kernel $k = \sum_{u=1}^{m} \beta_u k_u$ that jointly maximizes the test power and minimizes the Type II error.

We note that the DAN objective (4) is essentially a minimax problem; i.e., we compute $\min_{\Theta} \max_{\mathcal{K}} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right) \sigma_k^{-2}$. The CNN parameter Θ is learned by minimizing MK-MMD as a domain discrepancy, while the MK-MMD parameter $\boldsymbol{\beta}$ is learned by minimizing the Type II error. Both criteria are dedicated to an effective adaptation of domain discrepancy, aiming to consolidate the transferability of DAN features. We accordingly adopt an alternating optimization that updates Θ by mini-batch SGD (5) and $\boldsymbol{\beta}$ by QP (8) iteratively. Both updates cost $O(n)$ and are scalable to large datasets.

3.3. Analysis

We provide an analysis of the expected target-domain risk of our approach, making use of the theory of domain adaptation (Ben-David et al., 2007; 2010; Mansour et al., 2009) and the theory of kernel embedding of probability distributions (Sriperumbudur et al., 2009; Gretton et al., 2012a;b).

Theorem 1 *Let $\theta \in \mathcal{H}$ be a hypothesis, $\epsilon_s(\theta)$ and $\epsilon_t(\theta)$ be the expected risks of source and target respectively, then*

$$\epsilon_t(\theta) \leqslant \epsilon_s(\theta) + 2d_k(p, q) + C, \tag{9}$$

where C is a constant for the complexity of hypothesis space and the risk of an ideal hypothesis for both domains.

Proof sketch: A result from Ben-David et al. (2007) shows that $\epsilon_t(\theta) \leqslant \epsilon_s(\theta) + d_{\mathcal{H}}(p, q) + C_0$, where $d_{\mathcal{H}}(p, q)$ is the $\mathcal{H}$-divergence between p and q, which is defined as

$$d_{\mathcal{H}}(p, q) \triangleq 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x}^s \sim p}\left[\eta(\mathbf{x}^s) = 1\right] - \Pr_{\mathbf{x}^t \sim q}\left[\eta(\mathbf{x}^t) = 1\right] \right|. \tag{10}$$

The $\mathcal{H}$-divergence relies on the capacity of the hypothesis space $\mathcal{H}$ to distinguish distributions p from q, and $\eta \in \mathcal{H}$ can be viewed as a *two-sample* classifier. By choosing η as a (kernel) Parzen window classifier (Sriperumbudur et al., 2009), $d_{\mathcal{H}}(p, q)$ can be bounded by the empirical estimate

$$\begin{aligned} d_{\mathcal{H}}(p, q) & \leqslant \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + C_1 \\ & \leqslant 2\left(1 - \inf_{\eta \in \mathcal{H}}\left[\sum_{i=1}^{n_s} \frac{L\left[\eta(\mathbf{x}_i^s) = 1\right]}{n_s} + \sum_{j=1}^{n_t} \frac{L\left[\eta(\mathbf{x}_j^t) = -1\right]}{n_t}\right]\right) + C_1 \\ & = 2\left(1 + d_k(p, q)\right) + C_1, \end{aligned} \tag{11}$$

where $L(\cdot)$ is the linear loss function of the Parzen window classifier η, $L[\eta = 1] \triangleq -\eta$, $L[\eta = -1] \triangleq \eta$. By explicitly minimizing MK-MMD in multiple layers, the features and classifier learned by the proposed DAN model can decrease the upper bound on target risk. The source classifier and the two-sample classifier together provide a way to assess the adaptation performance, and can facilitate model selection. Note that we maximize MK-MMD w.r.t. $\boldsymbol{\beta}$ (7) to minimize Type II test error, and to help the Parzen window classifier achieve minimal risk of two-sample discrimination in (11).

4. Experiments

We compare the DAN model to state-of-the-art transfer learning and deep learning methods on both unsupervised and semi-supervised adaptation problems, focusing on the efficacy of multi-layer adaptation with multi-kernel MMD.

4.1. Setup

Office-31 (Saenko et al., 2010) This dataset is a standard benchmark for domain adaptation. It consists of 4,652 images within 31 categories collected from three distinct domains: *Amazon* (**A**), which contains images downloaded from `amazon.com`, *Webcam* (**W**) and *DSLR* (**D**), which are images taken by web camera and digital SLR camera in an office with different environment variation, respectively. We evaluate our method across the 3 transfer tasks, **A** → **W**, **D** → **W** and **W** → **D**, which are commonly adopted in deep learning methods (Donahue et al., 2014; Tzeng et al., 2014). For completeness, we further include the evaluation on the other 3 transfer tasks, **A** → **D**, **D** → **A** and **W** → **A**. **Office-10 + Caltech-10** (Gong et al., 2012). This dataset consists of the 10 common categories shared by the Office-31 and Caltech-256 (**C**) (Griffin et al., 2007) datasets and is widely adopted in transfer learning methods (Long et al., 2013; Baktashmotlagh et al., 2013). We can build another 6 transfer tasks: **A** → **C**, **W** → **C**, **D** → **C**, **C** → **A**, **C** → **W**, and **C** → **D**. With more transfer tasks, we are targeting an *unbiased* look at the dataset bias (Torralba & Efros, 2011).

We compare to a variety of methods: TCA (Pan et al., 2011), GFK (Gong et al., 2012), CNN (Krizhevsky et al., 2012), LapCNN (Weston et al., 2008), and DDC (Tzeng et al., 2014). Specifically, TCA is a conventional transfer learning method based on MMD-regularized PCA. GFK is a widely-adopted method for our datasets which interpolates across intermediate subspaces to bridge the source and target. CNN was the leading method in the ImageNet 2012 competition, and it turns out to be a strong model for learning transferable features (Yosinski et al., 2014). LapCNN is a semi-supervised variant of CNN based on Laplacian graph regularization. Finally, DDC is a domain adaptation variant of CNN that adds an adaptation layer between the $fc7$ and $fc8$ layers that is regularized by single-kernel MMD. We implement the CNN-based methods, i.e., CNN, LapCNN, DDC, and DAN based on the Caffe (Jia et al., 2014) implementation of AlexNet (Krizhevsky et al., 2012) trained on the ImageNet dataset. In order to study the efficacy of *multi-layer* adaptation and *multi-kernel* MMD, we evaluate several variants of DAN: (1) DAN using only one hidden layer, either $fc7$ or $fc8$ for adaptation, termed DAN_7 and DAN_8 respectively; (2) DAN using single-kernel MMD for adaptation, termed DAN_{SK}.

We mainly follow standard evaluation protocol for unsupervised adaptation and use all source examples with labels and all target examples without labels (Gong et al., 2013). To make our results directly comparable to most published results, we report a classical protocol (Saenko et al., 2010) in that we randomly down-sample the source examples, and further require 3 labeled target examples per category for semi-supervised adaptation. We compare the averages and standard errors of classification accuracy for each task. For baseline methods, we follow the standard procedures for model selection as explained in their respective papers. For MMD-based methods (i.e., TCA, DDC, and DAN), we use a Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\gamma}$ with the bandwidth γ set to the median pairwise distances on the training data—the *median heuristic* (Gretton et al., 2012b). We use multi-kernel MMD for DAN, and consider a family of m Gaussian kernels $\{k_u\}_{u=1}^m$ by varying bandwidth γ_u between $2^{-8}\gamma$ and $2^8\gamma$ with a multiplicative step-size of $2^{1/2}$ (Gretton et al., 2012b). As minimizing MMD is equivalent to maximizing the error of classifying the source from the target (two-sample classifier) (Sriperumbudur et al., 2009), we can automatically select the MMD penalty parameter λ on a validation set (comprised of source-labeled instances and target-unlabeled instances) by jointly assessing the test errors of the source classifier and the two-sample classifier. We use the fine-tuning architecture (Yosinski et al., 2014), however, due to limited training examples in our datasets, we fix convolutional layers $conv1$–$conv3$ that were copied from pre-trained model, fine-tune $conv4$–$conv5$ and fully connected layers $fc6$–$fc7$, and train classifier layer $fc8$, both via back propagation. As the classifier is trained from scratch, we set its learning rate to be 10 times that of the lower layers. We use stochastic gradient descent (SGD) with 0.9 momentum and the learning rate annealing strategy implemented in Caffe, and cross-validate base learning rate between 10^{-5} and 10^{-2} with a multiplicative step-size $10^{1/2}$.

4.2. Results and Discussion

The unsupervised adaptation results on the first six *Office-31* transfer tasks are shown in Table 1, and the results on the other six *Office-10 + Caltech-10* transfer tasks are shown in Table 2. To directly compare with DDC, we report semi-supervised adaptation results of the same tasks used by DDC in Table 3. We can observe that DAN significantly outperforms the comparison methods on most transfer tasks, and achieves comparable performance on the easy transfer tasks, **D** → **W** and **W** → **D**, where source and target are similar (Saenko et al., 2010). This is reasonable as the adaptability may vary across different transfer tasks. The performance boost demonstrates that our architecture of multi-layer adaptation via multi-kernel MMD is able to transfer pre-trained deep models across different domains.

From the experimental results, we can make the following observations. (1) Deep learning based methods significantly outperform conventional *shallow* transfer learning methods by a large margin. (2) Among the deep learning methods, the semi-supervised LapCNN provides no improvement over CNN, suggesting that the challenge of domain discrepancy cannot be readily bridged by semi-

Table 1. Accuracy on *Office-31* dataset with standard unsupervised adaptation protocol (Gong et al., 2013).

Method	A → W	D → W	W → D	A → D	D → A	W → A	Average
TCA	21.5 ± 0.0	50.1 ± 0.0	58.4 ± 0.0	11.4 ± 0.0	8.0 ± 0.0	14.6 ± 0.0	27.3
GFK	19.7 ± 0.0	49.7 ± 0.0	63.1 ± 0.0	10.6 ± 0.0	7.9 ± 0.0	15.8 ± 0.0	27.8
CNN	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
LapCNN	60.4 ± 0.3	94.7 ± 0.5	**99.1** ± 0.2	63.1 ± 0.6	51.6 ± 0.4	48.2 ± 0.5	69.5
DDC	61.8 ± 0.4	95.0 ± 0.5	98.5 ± 0.4	64.4 ± 0.3	52.1 ± 0.8	52.2 ± 0.4	70.6
DAN$_7$	63.2 ± 0.2	94.8 ± 0.4	98.9 ± 0.3	65.2 ± 0.4	52.3 ± 0.4	52.1 ± 0.4	71.1
DAN$_8$	63.8 ± 0.4	94.6 ± 0.5	98.8 ± 0.6	65.8 ± 0.4	52.8 ± 0.4	51.9 ± 0.5	71.3
DAN$_{\text{SK}}$	63.3 ± 0.3	95.6 ± 0.2	99.0 ± 0.4	65.9 ± 0.7	53.2 ± 0.5	52.1 ± 0.4	71.5
DAN	**68.5** ± 0.4	**96.0** ± 0.3	99.0 ± 0.2	**67.0** ± 0.4	**54.0** ± 0.4	**53.1** ± 0.3	**72.9**

Table 2. Accuracy on *Office-10* + *Caltech-10* dataset with standard unsupervised adaptation protocol (Gong et al., 2013).

Method	A → C	W → C	D → C	C → A	C → W	C → D	Average
TCA	42.7 ± 0.0	34.1 ± 0.0	35.4 ± 0.0	54.7 ± 0.0	50.5 ± 0.0	50.3 ± 0.0	44.6
GFK	41.4 ± 0.0	26.4 ± 0.0	36.4 ± 0.0	56.2 ± 0.0	43.7 ± 0.0	42.0 ± 0.0	41.0
CNN	83.8 ± 0.3	76.1 ± 0.5	80.8 ± 0.4	91.1 ± 0.2	83.1 ± 0.3	89.0 ± 0.3	84.0
LapCNN	83.6 ± 0.6	77.8 ± 0.5	80.6 ± 0.4	**92.1** ± 0.3	81.6 ± 0.4	87.8 ± 0.4	83.9
DDC	84.3 ± 0.5	76.9 ± 0.4	80.5 ± 0.2	91.3 ± 0.3	85.5 ± 0.3	89.1 ± 0.3	84.6
DAN$_7$	84.7 ± 0.3	78.2 ± 0.5	81.8 ± 0.3	91.6 ± 0.4	87.4 ± 0.3	88.9 ± 0.5	85.4
DAN$_8$	84.4 ± 0.3	80.8 ± 0.4	81.7 ± 0.2	91.7 ± 0.3	90.5 ± 0.4	89.1 ± 0.4	86.4
DAN$_{\text{SK}}$	84.1 ± 0.4	79.9 ± 0.4	81.1 ± 0.5	91.4 ± 0.3	86.9 ± 0.5	89.5 ± 0.3	85.5
DAN	**86.0** ± 0.5	**81.5** ± 0.3	**82.0** ± 0.4	92.0 ± 0.3	**92.0** ± 0.4	**90.5** ± 0.2	**87.3**

Table 3. Accuracy on *Office-31* dataset with classic unsupervised and semi-supervised adaptation protocols (Saenko et al., 2010).

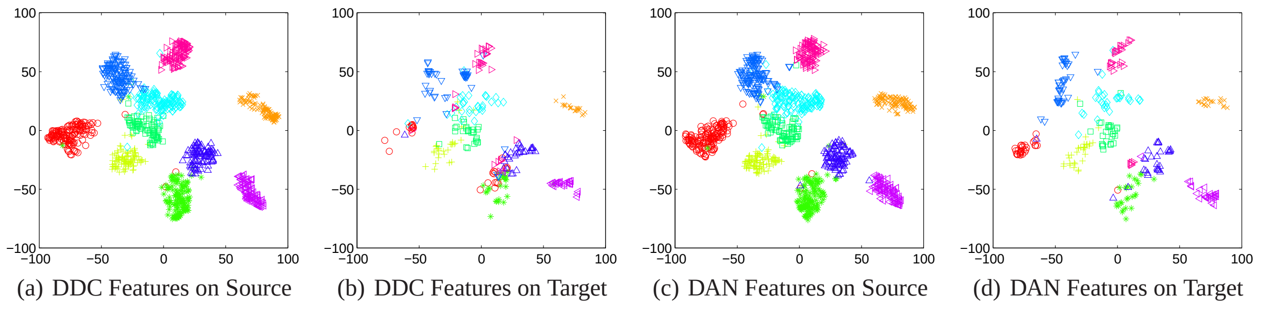
Method	A → W	D → W	W → D	Average
DDC	59.4 ± 0.8	92.5 ± 0.3	91.7 ± 0.8	81.2
DAN	**66.0** ± 0.4	**93.5** ± 0.2	**95.3** ± 0.3	**84.9**
DDC	84.1 ± 0.6	95.4 ± 0.4	96.3 ± 0.3	91.9
DAN	**85.7** ± 0.3	**97.2** ± 0.2	**96.4** ± 0.2	**93.1**

supervised learning. (3) DDC, a cross-domain variant of CNN with single-layer adaptation via single-kernel MMD, generally outperforms CNN, confirming its effectiveness in learning transferable features using domain-adaptive deep models. Note that while DDC based on Caffe AlexNet was shown to significantly outperform DeCAF (Donahue et al., 2014) in which fine-tuning was not carried out, it does not yield a large gain over Caffe AlexNet using fine-tuning. This shows the limitation of single-layer adaptation via single-kernel MMD, which cannot explore the strengths of deep networks and multiple kernels for domain adaptation.

To dive deeper into DAN, we present the results of three variants of DAN: (1) DAN$_7$ and DAN$_8$ achieve better accuracy than DDC, which highlights that multi-kernel MMD can bridge the domain discrepancy more effectively than single-kernel MMD. The reason is that multiple kernels with different bandwidths can match both the low-order moments and high-order moments to minimize the Type II error (Gretton et al., 2012b). (2) DAN$_{\text{SK}}$ also attains higher accuracy than DDC, which confirms the capability of deep architecture for distribution adaptation. The rationale is similar to that of deep networks: each layer of deep network is intended to extract features at a different abstraction level, and hence we need to match the distributions at each task-specific layer to consolidate the adaptation quality at all levels. The multi-layer architecture is one of the most critical contributors to the efficacy of deep learning, and we believe it is also important for MMD-based adaptation. The evidence of comparable performance between the multi-layer variant DAN$_{\text{SK}}$ and multi-kernel variants DAN$_7$ and DAN$_8$ shows their equal importance for domain adaptation. As expected, DAN obtains the best performance by jointly exploring multi-layer adaptation with multi-kernel MMD. Another benefit of DAN is that it uses a linear-time unbiased estimate of the kernel embedding, which makes it an order more efficient than existing methods TCA and DDC. Though Tzeng et al. (2014) speed up DDC by computing the MMD within each mini-batch of the SGD, this leads to a biased estimate of MMD and lower adaptation accuracy.

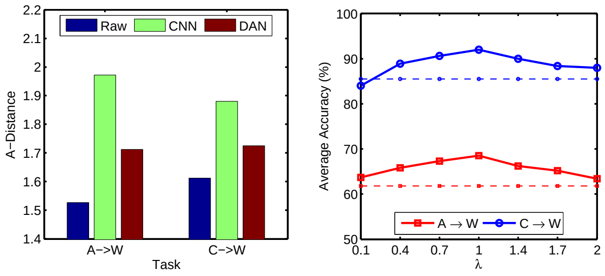
4.3. Empirical Analysis

Feature Visualization To demonstrate the transferability of the DAN learned features, we follow Donahue et al. (2014) and Tzeng et al. (2014) and plot in Figures 2(a)–2(b) and 2(c)–2(d) the t-SNE embeddings of the images



(a) DDC Features on Source (b) DDC Features on Target (c) DAN Features on Source (d) DAN Features on Target

Figure 2. Feature visualization: t-SNE of DDC features on source (a) and target (b); t-SNE of DAN features on source (c) and target (d).



(a) $\mathcal{A}$-Distance (b) Accuracy vs. λ

Figure 3. Empirical analysis: (a) $\mathcal{A}$-Distance of CNN & DAN features; (b) sensitivity of λ (dashed lines show best baseline results).

in task $\mathbf{C} \to \mathbf{W}$ with DDC features and DAN features, respectively. We make the following observations: (1) With DDC features, the target points are not discriminated very well, while with DAN features, the points are discriminated much better. (2) With DDC features, the categories between the source and the target are not aligned very well, while with DAN features, the categories are aligned much better between domains. Both these observations can explain the superior performance of DAN over DDC: (1) implies that the target points are more easily discriminated with DAN features, and (2) implies that the target points can be better discriminated with the source classifier. DAN can learn more transferable features for effective domain adaptation.

$\mathcal{A}$-Distance A theoretical result in Ben-David et al. (2010) suggests $\mathcal{A}$-distance as a measure of domain discrepancy. As computing the exact $\mathcal{A}$-distance is intractable, an approximate distance is defined as $\hat{d}_{\mathcal{A}} = 2\,(1 - 2\epsilon)$, where ϵ is the generalization error of a two-sample classifier (kernel SVM in our case) trained on the binary problem to distinguish input samples between the source and target domains. Figure 3(a) displays $\hat{d}_{\mathcal{A}}$ on transfer tasks $\mathbf{A} \to \mathbf{W}$ and $\mathbf{C} \to \mathbf{W}$ using Raw features, CNN features, and DAN features, respectively. It reveals a surprising observation that the $\hat{d}_{\mathcal{A}}$ on both CNN and DAN features are larger than the $\hat{d}_{\mathcal{A}}$ on Raw features. This implies that abstract deep features can be salient both for discriminating different categories and different domains, which is consistent with Glorot et al. (2011). However, domain adaptation may be deteriorated by the enlarged domain discrepancy (Ben-David et al., 2010). It is desirable that $\hat{d}_{\mathcal{A}}$ on DAN feature is smaller than $\hat{d}_{\mathcal{A}}$ on CNN feature, which guarantees more transferable features.

Parameter Sensitivity We investigate the effects of the parameter λ. Figure 3(b) gives an illustration of the variation of transfer classification performance as $\lambda \in \{0.1, 0.4, 0.7, 1, 1.4, 1.7, 2\}$ on tasks $\mathbf{A} \to \mathbf{W}$ and $\mathbf{C} \to \mathbf{W}$. We can observe that the DAN accuracy first increases and then decreases as λ varies and demonstrates a bell-shaped curve. This confirms the motivation of jointly learning deep features and adapting distribution discrepancy, since a good trade-off between them can enhance feature transferability.

5. Conclusion

In this paper, we have proposed a novel Deep Adaptation Network (DAN) architecture to enhance the transferability of features from task-specific layers of the neural network. We confirm that while general features can generalize well to a novel task, specific features tailored to an original task cannot bridge the domain discrepancy effectively. We show that feature transferability can be enhanced substantially by mean-embedding matching of the multi-layer representations across domains in a reproducing kernel Hilbert space. An optimal multi-kernel selection strategy further improves the embedding matching effectiveness, while an unbiased estimate of the mean embedding naturally leads to a linear-time algorithm that is very desirable for deep learning from large-scale datasets. An extensive empirical evaluation on standard domain adaptation benchmarks demonstrates the efficacy of the proposed model against previous methods.

As deep features transition from general to specific along the network, it is interesting to study the principled way of deciding the boundary of generality and specificity, and the application of distribution adaptation to the convolutional layers of CNN to further enhance the feature transferability.

Acknowledgments

This work was supported by the National Science Funds for Distinguished Young Scholars (No. 613250154), National Science and Technology Supporting Program Project (No. 2015BAH14F02), and Tsinghua TNList Fund for Big Data Science and Technology.

References

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Marchand, M. Domain-adversarial neural networks. In *NIPS 2014 Workshop on Transfer and Multi-task learning: Theory Meets Practice*, 2014.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, 2013.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. *NIPS*, 2007.

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*, 35(8):1798–1828, 2013.

Chen, M., Xu, Z., Weinberger, K. Q., and Sha, F. Marginalized denoising autoencoders for domain adaptation. In *ICML*, 2012.

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 *ICML*, 2014.

Ge, L., Gao, J., Li, X., and Zhang, A. Multi-source deep learning for information trustworthiness estimation. In *KDD*, 2013.

Ghifary, M., Kleijn, W. B., and Zhang, M. Domain adaptive neural networks for object recognition. Technical report, arXiv:1409.6041, 2014.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

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

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

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

Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., and Fukumizu, K. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, 2012b.

Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv:1207.0580, 2012.

Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R., Darrell, T., and Saenko, K. LSDA: Large scale detection through adaptation. In *NIPS*, 2014.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. *COLT*, 2009.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal deep learning. In *ICML*, 2011.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 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 and Learning Systems*, 22(2):199–210, 2011.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. Technical report, arXiv:1409.0575, 2014.

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

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Lanckriet, G., and Schölkopf, B. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *NIPS*, 2009.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR*, 2011.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. Technical report, arXiv:1412.3474, 2014.

Wang, X. and Schneider, J. Flexible transfer learning under support and model shift. In *NIPS*, 2014.

Weston, J., Rattle, F., and Collobert, R. Deep learning via semi-supervised embedding. In *ICML*, 2008.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *NIPS*, 2014.

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