Learning Transferable Visual Features with Very Deep Adaptation Networks

Mingsheng Long, Zhangjie Cao, Jianmin Wang, Han Zhu, and Michael I. Jordan

Abstract—Domain adaptation generalizes a learning machine across source domain and target domain under different distributions. Recent studies reveal that deep neural networks can learn transferable features that generalize well to similar novel tasks for domain adaptation. However, as deep features eventually transition from general to specific along the network, feature transferability drops significantly in higher task-specific layers with increasing domain discrepancy. To formally reduce the dataset bias and enhance the feature transferability in task-specific layers, this paper presents a novel framework for deep adaptation networks, which generalizes deep convolutional neural networks to domain adaptation. The framework embeds the deep features of all task-specific layers to reproducing kernel Hilbert spaces (RKHSs) and optimally match different domain distributions. The deep features are made more transferable by exploring very deep architectures and low-density separation of target-unlabeled data, while the domain discrepancy is further reduced using multiple kernel learning and maximal testing power for kernel embedding matching. The framework can learn transferable visual features with statistical guarantees, and can scale linearly using unbiased estimate of kernel embedding. Extensive empirical evidence shows that the proposed networks yield state of the art results on standard visual domain adaptation benchmarks.

Index Terms—Domain adaptation, deep learning, convolutional neural network, two-sample test, multiple kernel learning.

1 Introduction

The generalization error of supervised learning machine with limited training examples will be uncontrollable, while manual labeling of sufficient training data for diverse application domains may be prohibitive. In these scenarios, there is strong incentive to establishing effective algorithms to reduce the labeling consumption, typically by leveraging off-the-shelf labeled data from related source domain to the target domain. Domain adaptation addresses the problem that we have data from two related domains but they follow 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 pose, occlusion, or illumination. Domain adaptation enables knowledge transfer from labeled source domain to unlabeled target domain by exploring domain-invariant structures that bridge different domains under substantial distribution discrepancy [1], [2].

Among the approaches to knowledge transfer, an important strategy 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 [3], [4], [5], [6], [7]. However, the recent studies have shown that deep neural networks can learn much more transferable features for domain adaptation [8], [9], [10], 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, hence making features robust to noises.

While deep networks are more powerful for learning generally transferable features, the latest findings also reveal that the deep features must eventually transition from general to specific along the network, and the feature transferability drops substantially in higher task-specific layers with increasing domain discrepancy. In other words, the features in the higher layers of the deep network will depend greatly on specific dataset and task [10], which are task-specific features and are not safely transferable to novel tasks. Another curious phenomenon is that disentangling the exploratory factors of variations in the higher task-specific layers of deep networks may further increase the domain discrepancy, as different domains under the new deep features will become more mutually distinguishable [8]. Although deep features are discriminative for classification, increased dataset bias may deteriorate domain adaptation performance, leading to statistically *unbounded* error for the target task [2], [11], [12].

This paper is motivated by the literature's latest understanding on the transferability of deep neural networks [10]. We propose a new framework of deep adaptation networks, which generalizes deep convolutional neural networks to the domain adaptation scenario. The main idea is to formally reduce the dataset bias and enhance the feature transferability in task-specific layers of the deep neural networks. To establish this goal, the deep features of all task-specific higher layers are embedded to reproducing kernel Hilbert spaces (RKHSs) in which the kernel embeddings of different domain distributions are matched statistically. Since kernel embedding matching is sensitive to the kernel

- *M. Long, Z. Cao, J. Wang, and H. Zhu are with the School of Software, Tsinghua TNList Laboratory and NEL-BDSS Laboratory, Tsinghua University, China. E-mail: {mingsheng, jimwang}@tsinghua.edu.cn, {caozj14, zhuhan14}@mails.tsinghua.edu.cn. Corresponding author: J. Wang.*
- *M. I. Jordan is with the Department of EECS and the Department of Statistics, University of California, Berkeley. E-mail: jordan@berkeley.edu.*

Manuscript received January, 2016; revised April, 2017.

selections, an optimal multi-kernel learning method [13] and a maximal testing power method [14] are explored to further reduce the domain discrepancy. The deep features are made more transferable by exploring very deep GoogLeNet [15] and ResNet [16], and the low-density separation criterion for target-unlabeled data [17]. We implement a linear-time unbiased estimate of the kernel mean embedding to enable scalable training, which is very desirable for deep learning. Finally, since deep models pre-trained on large repositories such as ImageNet [18] are representative and effective for general perception tasks [10], [19], [20], [21], the proposed deep adaptation networks are trained by fine-tuning from AlexNet [22], GoogLeNet [15] and ResNet [16] pre-trained on ImageNet by Caffe [23] framework. Extensive empirical evidence shows that the proposed models yield state of the art results on the standard domain adaptation benchmarks.

This work makes substantial extension to our conference paper [24]. The main contributions are summarized as follows. **(1)** We propose a novel framework of deep adaptation networks for visual domain adaptation, where *multiple* task-specific layers are adapted, hence benefiting from "deep adaptation." **(2)** We explore *multiple* kernels for matching deep representations across domains, hence benefiting from "optimal matching." **(3)** We further learn *distinguishable* test locations for a new two-sample test to maximize the distinguishability of distributions, hence benefiting from "adversarial matching." **(4)** We exploit the *low-density separation* criterion by entropy minimization on the target-unlabeled data, hence benefiting from "semi-supervised adaptation." **(5)** We go deeper with feature transferability of the *very deep* GoogLeNet, hence benefiting from "very deep adaptation."

2 Related Work

This work is related to transfer learning [1], which builds models that can bridge different domains or tasks, explicitly taking the cross-domain discrepancy into account. Transfer learning is to mitigate the burden of manual labeling for machine learning [3], [5], [6], [25], [7], [26], computer vision [27], [28], [29], [4], [20], [30] and natural language processing [31], [32]. It is a consensus that cross-domain discrepancy in probability distributions of different domains should be formally reduced. The major bottleneck is how to match different domain distributions effectively. Most existing methods learn a new shallow representation model by which the domain discrepancy can be explicitly minimized. However, without learning deep features which can suppress domain-specific exploratory factors of variations, the transferability of shallow features is restricted by task-specific structures. There are several very recent attempts in learning domain-invariant features in the context of shallow networks [33], but these proposals generally underperform deep networks.

Deep neural networks can learn abstract representations that disentangle and hide some different explanatory factors of variations behind data samples [34]. The learned deep representations manifest invariant factors underlying different populations and are transferable from the original tasks to similar novel tasks [10]. Therefore, deep neural networks have been well explored for domain adaptation [8], [19], [20], multimodal and multi-task learning problems [31], [35], where significant performance gains have been witnessed. These methods depend on the mild assumption that deep neural networks can learn the desired invariant representations that are transferable across different tasks. In reality, the domain discrepancy can be reduced, but not removed, by deep networks [8], [36]. Dataset shift has posed a bottleneck to the transferability of deep features, resulting in statistically *unbounded* risk for target tasks [11], [2], [12].

This work is primarily motivated by Yosinski et al. [10], which comprehensively quantifies feature transferability of deep convolutional neural networks. The method thereof 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 new target task, hence the method cannot be directly applicable to domain adaptation. Nonetheless, it reveals that deep features must eventually transition from general to specific along the network, and feature transferability drops substantially in multiple higher layers, and this is the main problem addressed by this work.

The domain discrepancy should be formally reduced to achieve lower transfer errors [11], [12], [2]. Several parallel works [36], [37], [30] extend deep convolutional networks (CNN) to domain adaptation either by adding an adaptation layer through which the means of distributions are matched [36], or by adding a fully connected subnetwork as a domain discriminator while the deep features are learned to confuse the domain discriminator in a domain-adversarial training paradigm [37], [30]. While performance was improved, these state of the art methods may be restricted by several limitations. **(1)** These methods adapt only one layer of the deep network, while there are multiple task-specific layers where the features are not safely transferable [10]. **(2)** They add new adaptation layers or new parametric subnetwork, which make them difficult to train with limited data [30], while limited labeled data is a common scenario of domain adaptation. **(3)** They do not fully exploit the target data to refine the source classifier, hence the source classifier may still be misspecified for the target data. **(4)** The transferability with very deep networks such as GoogLeNet [15] and ResNet [16] has not been explored by the previous methods.

3 Very Deep Domain Adaptation Networks

This section presents two deep architectures with different network-depths for learning transferable visual features: *deep adaptation network* (DAN) based on the breakthrough AlexNet [22], and *very deep adaptation network* (VDAN) based on the state of the art GoogLeNet [15]. By going deeper with convolutional networks, we can approach a deeper understanding and consolidation for deep-feature transferability, which will enable more effective visual domain adaptation.

In domain adaptation problems, we are provided with a *source* domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a *target* domain $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ of n_t examples, where the target domain $\mathcal{D}_t = \mathcal{D}_l \cup \mathcal{D}_u$ may contain both labeled data $\mathcal{D}_l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{n_l}$ and unlabeled data $\mathcal{D}_u = \{\mathbf{x}_i^u\}_{i=1}^{n_u}$, while $n_l \ll n_s$. Denote by $\mathcal{D}_a = \{(\mathbf{x}_i^a, y_i^a)\}_{i=1}^{n_a}$ all available labeled examples of the source and target domains, $\mathcal{D}_a = \mathcal{D}_s \cup \mathcal{D}_l$. We consider two common scenarios: *unsupervised* domain adaptation if the target domain is fully unlabeled ($\mathcal{D}_l = \emptyset$)

and *semi-supervised* domain adaptation if the target domain is partially labeled ($\mathcal{D}_l \neq \emptyset$). The source domain and target domain are sampled from probability distributions p and q respectively, and note $p \neq q$. The goal of this paper is to craft a deep neural network $y = f(\mathbf{x})$ which enables learning of transferable features to bridge the source-target discrepancy, such that the target risk $R_t(f) = \Pr_{(\mathbf{x},y)\sim q}[f(\mathbf{x}) \neq y]$ can be minimized by leveraging the source domain supervision.

3.1 Two-Sample Test Statistics

The challenge of domain adaptation mainly arises in that the target domain has no or only limited labeled information. To approach this problem, many existing methods aimed to bound the target error by the source error plus a discrepancy metric between source and target distributions p and q [12]. Two classes of statistics have been explored for *two-sample* testing and distribution comparison, which accepts or rejects the null hypothesis $p = q$ based on two samples generated from p and q: *Maximum Mean Discrepancy* (MMD) [38] and *Mean Embedding Test* (ME) [14], [39]. In this paper, we will explore these two test statistics for deep domain adaptation.

3.1.1 Generalized MMD Test

First, we describe the *generalized* MMD [40], [13], [41], which jointly maximizes the two-sample test power and minimizes the Type-II error, i.e. failure of rejecting false null hypothesis $p = q$. Technically, minimizing the Type-II error is reduced to maximizing a normalized counterpart of MMD between domains [13], which will maximize the distinguishability of data distributions to enable effective distribution matching.

Let $\mathcal{H}_k$ be the reproducing kernel Hilbert space (RKHS) induced with a *characteristic* kernel k. Then the *kernel mean embedding* of distribution p in $\mathcal{H}_k$ is a unique element $\mu_k(p)$ such that the expectation $\mathbb{E}_{\mathbf{x}\sim p} f(\mathbf{x}) = \langle f(\mathbf{x}), \mu_k(p) \rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$. That is, all important information conveyed in distribution p is encoded into the embedding $\mu_k(p)$ so that we can learn through $\mu_k(p)$ instead of p, which removes the necessity of density estimation of p. The multi-kernel maximum mean discrepancy (MK-MMD) between distributions p and q is defined as the RKHS-distance between the kernel mean embeddings of p and q, and the squared MK-MMD is

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

where $\phi(\cdot)$ is a nonlinear feature mapping that induces $\mathcal{H}_k$. Given $\mathcal{D}_s$ and $\mathcal{D}_t$ as the sets of samples from distributions p and q respectively, the empirical estimate of MK-MMD is

$$\begin{aligned} M_k(\mathcal{D}_s, \mathcal{D}_t) \triangleq & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\mathbf{x}_i^s, \mathbf{x}_j^s) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(\mathbf{x}_i^t, \mathbf{x}_j^t) \\ & - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(\mathbf{x}_i^s, \mathbf{x}_j^t). \end{aligned} \tag{2}$$

The most important property is that $p = q$ iff $M_k(p,q) = 0$ [38]. The *characteristic* kernel $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ is defined as the convex combination of m PSD kernels $\{k_u\}$,

$$\mathcal{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geqslant 0, \forall u \right\}, \tag{3}$$

where the constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that the composed multi-kernel k is characteristic. As theoretically studied in [13], the kernel adopted for mean embeddings of p and q is critical in ensuring high test power and low Type-II test error, i.e. minimizing the probability of degenerated two-sample tests $M_k(p,q) \to 0$ when $p \neq q$. The kernel mean embeddings by different kernels $\{k_u\}$ with different bandwidths can characterize the distributions at different scales and thus match different orders of moments. By minimizing the Type-II error, we can automatically learn an optimal kernel for cross-domain distribution adaptation.

3.1.2 ME Test

The ME test [14] is a form of Hotellings T-squared statistic to compare distributions. The ME test is formally defined as

$$M_k(\mathcal{D}_s, \mathcal{D}_t) \triangleq n \bar{\mathbf{z}}_n^{\mathsf{T}} \mathbf{S}_n^{-1} \bar{\mathbf{z}}_n, \tag{4}$$

where $\bar{\mathbf{z}}_n \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i$, $\mathbf{S}_n \triangleq \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{z}_i - \bar{\mathbf{z}}_n)(\mathbf{z}_i - \bar{\mathbf{z}}_n)^{\mathsf{T}}$, and $\mathbf{z}_i \triangleq (k(\mathbf{x}_i^s, \mathbf{v}_j) - k(\mathbf{x}_i^t, \mathbf{v}_j))_{j=1}^{J} \in \mathbb{R}^J$. The J test locations $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^{J} \subset \mathbb{R}^J$ are chosen by $\max_{\mathcal{V}} n \bar{\mathbf{z}}_n^{\mathsf{T}} \mathbf{S}_n^{-1} \bar{\mathbf{z}}_n$, which maximizes ME test power and results in parsimonious and interpretable indication of how and where two distributions differ locally [39]. In this paper, we adopt unnormalized ME test, i.e. without $\mathbf{S}_n^{-1}$, which is shown by [14] to be a metric on the space of probability measures for any $\mathcal{V}$ and behave similarly for two-sample testing as the normalized ME in (4). To compare domains with different sizes n_s and n_t, we need to sample the target domain such that the number n of points is the same for source and target to perform ME test. It has been shown that ME test can be computed efficiently in linear time and has greater test power than MMD [39].

A successful strategy to control the domain discrepancy is to find an invariant feature representation through which the source domain and target domain are made similar [2], [11], [12]. MMD has been extensively explored in this line of works [3], [26], [4], [6], [7], but to date there has been no attempt for learning transferable visual features via MK-MMD or ME in deep networks. Hence, previous shallow transfer-learning methods may be restricted by weak representation power [34], while previous standard deep-learning methods may be restricted by weak adaptation efficacy [8], [36], [13].

3.2 Deep Adaptation Network

Deep learning [34] is equipped with the ability of learning distributed, compositional, and abstract representations for natural data such as image and speech. In this paper, we will explore the idea of MK-MMD and ME based distribution matching in deep networks for learning transferable visual features. We start with deep convolutional neural networks (CNN) [22], a strong model for learning high-quality visual features that are adaptable to new tasks [19], [9], [20]. The main challenge resides in that the target domain contain no or very limited labeled information, hence directly adapting CNN to the target domain via fine-tuning [19] is impossible or is prone to over-fitting. With the idea of domain adaptation, we are crafting 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 breakthrough AlexNet architecture [22], which comprises of five convolutional layers ($conv1$–$conv5$) and three fully connected layers ($fc6$–$fc8$), while $conv1$–$fc7$

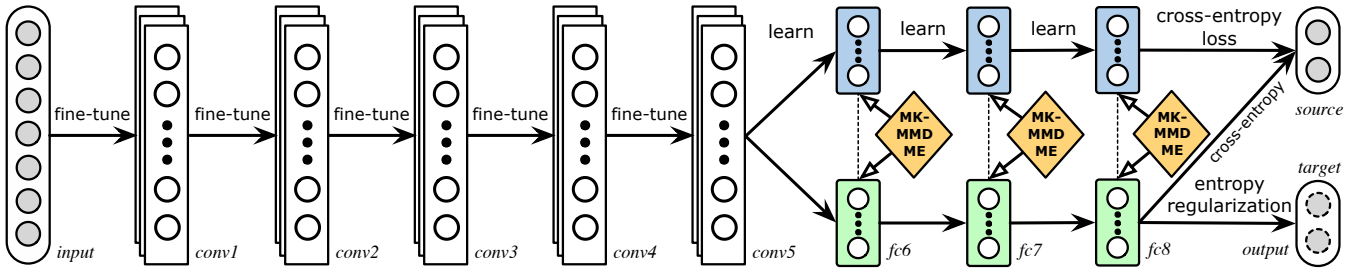


Fig. 1. The *deep adaptation network* (DAN) based on AlexNet [22]. Since deep features transition from general to specific along the network: (1) the features extracted by convolutional layers $conv1$–$conv5$ are transferable, hence these layers are learned via fine-tuning; (2) fully connected layers $fc6$–$fc8$ are tailored to fit task-specific structures, hence they are not safely transferable and should be adapted with MK-MMD/ME minimization. To mitigate the misspecification of the source classifier to the target data, the low-density separation of target data is explored by entropy minimization.

are feature layers and $fc8$ is the classifier layer. Specifically, each fully connected layer ℓ will learn a nonlinear mapping $\mathbf{h}_i^\ell = a^\ell(\mathbf{W}^\ell \mathbf{h}_i^{\ell-1} + \mathbf{b}^\ell)$, where $\mathbf{h}_i^\ell$ is the ℓ^{th}-layer hidden representation of example $\mathbf{x}_i$, $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ are the ℓ^{th}-layer weight and bias parameters, and a^ℓ is the ℓ^{th}-layer activation function, taken as rectifier units (ReLU) $a^\ell(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x})$ for the hidden layers or softmax units $a^\ell(\mathbf{x}) = e^{\mathbf{x}} / \sum_{j=1}^{|\mathbf{x}|} e^{x_j}$ for the output layer. Denote by $\mathcal{F} = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{\ell=1}^{l}$ the set of network parameters (l layers in total). The empirical error $E(\mathcal{D}_a; f)$ of CNN f on labeled data $\mathcal{D}_a = \{(\mathbf{x}_i^a, y_i^a)\}_{i=1}^{n_a}$ is

$$\min_{f \in \mathcal{F}} E(\mathcal{D}_a; f) = \frac{1}{n_a} \sum_{i=1}^{n_a} L(f(\mathbf{x}_i^a), y_i^a), \tag{5}$$

where $f(\mathbf{x}_i^a)$ is the conditional probability that CNN assigns point $\mathbf{x}_i^a$ to all labels, $L(\cdot, \cdot)$ is the cross-entropy loss function defined as $L(f(\mathbf{x}_i^a), y_i^a) = -\sum_{j=1}^{c} 1\{y_i^a = j\} \log f_j(\mathbf{x}_i^a)$, c is the number of classes, and $f_j(\mathbf{x}_i^a) = e^{h_{ij}^{a,l}} / \sum_{j'} e^{h_{ij'}^{a,l}}$ is the softmax function defined on the l^{th}-layer representation $\mathbf{h}_i^{a,l}$ that computes the probability of predicting point $\mathbf{x}_i^a$ to class j. Based on the quantification study of feature transferability [10], the convolutional layers can learn generic features which are transferable in layers $conv1$–$conv3$ and slightly specific in layers $conv4$–$conv5$ [10]. Hence, when adapting the pre-trained AlexNet model from the source domain to the target domain, we opt to fine-tune $conv1$–$conv5$ such that the efficacy of feature co-adaptation can be maximally preserved. Since we will perform distribution matching only for fully connected layers and pooling layers, we will not elaborate computational details of convolutional layers.

The literature has shown that the deep features learned by CNNs can disentangle the exploratory factors of variations underlying data distributions and facilitate knowledge transfer [19], [34]. However, the latest literature findings also reveal that the deep features can reduce, but not remove, the cross-domain distribution discrepancy [10], [36]. The deep features in standard CNNs must eventually transition from general to specific along the network, and the transferability gap grows with the cross-domain discrepancy and becomes particularly large when transferring the higher layers $fc6$–$fc8$ [10]. In other words, the fc layers are tailored to fit their original task at the expense of degraded performance on the target task, hence they cannot be directly transferred to the target domain by fine-tuning with little target supervision. In this paper, we fine-tune CNN on labeled examples and require the distributions of source and target to become similar under hidden representations of fully connected layers $\mathcal{L} = \{fc6, fc7, fc8\}$. This is implemented by minimizing a *multi-layer* MK-MMD or ME penalty that substitutes hidden representations $\{\mathbf{h}_i^\ell\}_{\ell \in \mathcal{L}}$ into MK-MMD (2) or ME-test (4) as

$$\min_{f \in \mathcal{F}} \max_{k \in \mathcal{K}} E(\mathcal{D}_s, \mathcal{D}_t; f, k) = \sum_{\ell \in \mathcal{L}} M_k\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right), \tag{6}$$

where $\mathcal{D}_s^\ell = \{\mathbf{h}_i^{s\ell}\}$ and $\mathcal{D}_t^\ell = \{\mathbf{h}_i^{t\ell}\}$ are the ℓ^{th}-layer hidden representations of the source and target points respectively, and $M_k(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell)$ is the MK-MMD/ME between source and target evaluated using the ℓ^{th}-layer hidden representations. $\mathcal{L}$ is the indices of layers where MMD/ME penalty is active, whose configuration is relying on the size of training data and the number of parameters to be fined-tuned or learned. It is notable that maximizing MK-MMD/ME with respect to kernel $k \in \mathcal{K}$ is equivalent to maximizing the two-sample test power when the test statistics (1) and (4) have isotropic covariance structure [40], [13]. This enables deep features to minimize an upper bound of the cross-domain discrepancy, which guarantees the goodness of distribution matching for domain adaptation. We will also consider the non-isotropic covariance structure when deriving algorithms in Section 4.

Finally, as it is very difficult to achieve zero cross-domain distribution discrepancy, i.e. MK-MMD/ME in Equation (6) do not reach zero, the source and target domains may still be somewhat different after optimizing Equations (5) and (6). In this case, the deep network classifier f still fails to classify the target-unlabeled data accurately. Due to the distribution discrepancy, deep classifier f must be able to pass through the low-density regions of the target-unlabeled data in order to perform well on the target domain. From the perspective of semi-supervised learning, the target-unlabeled data will be informative to the source classifier f because the source and target are different in distributions [42]. Different from semi-supervised learning where the labeled data are limited, in domain adaptation we have rich source-labeled data but they are not identically distributed with the target-unlabeled data. In this paper, for the first time, we exploit the *entropy minimization* principle [17] in the deep adaptation network, which favors the low-density separation between classes by minimizing the conditional-entropy $E(\mathcal{D}_u; f)$ of the class probability $p(y_i^u = j | \mathbf{x}_i^u; f) = f_j(\mathbf{x}_i^u)$ on unlabeled data $\mathcal{D}_u$:

$$\min_{f \in \mathcal{F}} E(\mathcal{D}_u; f) = \frac{1}{n_u} \sum_{i=1}^{n_u} H(f(\mathbf{x}_i^u)), \tag{7}$$

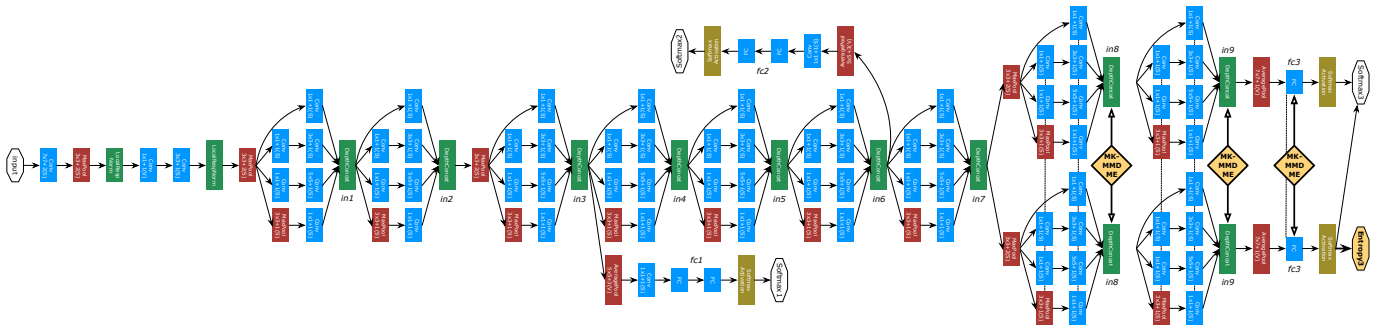


Fig. 2. The *very deep adaptation network* (VDAN) based on GoogLeNet [15]. Since deep features transition from general to specific along the network: (1) the features by inception layers $in1$–$in7$ are transferable, hence these layers are learned via fine-tuning; (2) inception layers $in8$–$in9$ and fully connected layer $fc3$ are tailored to fit task-specific structures, hence they are not safely transferable and are adapted with MK-MMD/ME. To mitigate the misspecification of the source classifier to the target data, the low-density separation of target data is explored by entropy minimization.

where $f\left(\mathbf{x}_i^u\right)$ is the conditional probability that the CNN assigns unlabeled-point $\mathbf{x}_i^u$ to all classes, and $H(\cdot)$ is the conditional-entropy loss function defined as $H\left(f\left(\mathbf{x}_i^u\right)\right) = -\sum_{j=1}^{c} f_j\left(\mathbf{x}_i^u\right) \log f_j\left(\mathbf{x}_i^u\right)$, c is the number of classes, and $f_j\left(\mathbf{x}_i^u\right)$ is the probability of predicting point $\mathbf{x}_i^u$ to class j. By minimizing the entropy penalty (7), deep network classifier f is made directly accessible to the target-unlabeled data and will adjust itself to pass through the low-density regions of target domain. Along with MMD/ME minimization (6), this will make deep network classifier f better fit target data.

Based on the aforementioned analysis, the invariance of deep features, the adaptation of domain distributions, and the separation of low-density regions all contribute critically to the domain adaptation performance. Hence, we integrate (5), (6) and (7) in a unified *deep adaptation network* (DAN) as

$$\begin{aligned}\min_{f\in\mathcal{F}} \max_{k\in\mathcal{K}} \; & E\left(\mathcal{D}_a; f\right) + \gamma E\left(\mathcal{D}_u; f\right) + \lambda E\left(\mathcal{D}_s, \mathcal{D}_t; f, k\right) \\ & = \frac{1}{n_a}\sum_{i=1}^{n_a} L\left(f\left(\mathbf{x}_i^a\right), y_i^a\right) + \frac{\gamma}{n_u}\sum_{i=1}^{n_u} H\left(f\left(\mathbf{x}_i^u\right)\right) \\ & + \lambda \sum_{\ell\in\mathcal{L}} M_k\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right),\end{aligned} \tag{8}$$

where γ and λ are the tradeoff parameters respectively for the conditional-entropy penalty (7) and the multi-layer MK-MMD/ME penalty (6), while $1/n_a$ and $1/n_u$ help make the three error terms more balanced. As training deep CNNs requires a large amount of labeled data that is prohibitive for many domain adaptation applications, we start with the AlexNet model pre-trained on ImageNet 2012 data and fine-tune it as in [10], [36]. With the proposed DAN model (8), we are able to learn transferable features that are both discriminative due to CNN and entropy minimization, and domain-unbiased benefiting from MK-MMD or ME minimization.

3.3 Very Deep Adaptation Network

The composition of multiple levels of nonlinearity in neural networks is key to efficiently model complex relationships across exploratory factors and to enable better generalization performance on challenging perception tasks [22], [34]. By going even deeper with convolutions, the very deep convolutional networks including GoogLeNet [15] and ResNet [16] have made breakthroughs in achieving new state of the art results in ImageNet Large-Scale Visual Recognition Challenge 2015 (ILSVRC14) [18]. Although the transferability of AlexNet features has been extensively quantified [10] and enhanced by the proposed DAN model, it still remains unclear whether very deep neural networks can learn more transferable features and how the feature transferability may change with the depths of very deep networks. In this paper, we approach this goal by extending deep adaptation network to the very deep networks, leading to the *very deep adaptation network* (VDAN) for learning transferable visual features. Figure 2 gives an illustration of the VDAN model.

GoogLeNet: We extend the GoogLeNet architecture [15], which is comprised of nine *inception* layers ($in1$–$in9$) and three fully connected layers ($fc1$–$fc3$) for three deeply supervised softmax classifiers, which are plugged in layers $in3$, $in6$ and $in9$ to mitigate the gradient vanishing trap. As it is difficult to add penalties to the convolutional layers, we opt to add the *multi-layer* MK-MMD/ME penalty on fully connected layers (fc), pooling layers (pl), and concat layers (cc). Denote by $cc\ell$ the concat layer of the ℓ^{th} inception layer, which is the concatenation of the outputs of all convolution branches of the inception layer and is ready for penalization. As Figure 2 shows, for the very deep adaptation model in Equation (8), we specify the last softmax classifier $fc3$ as the deep network classifier f for the cross-entropy loss (5) and the conditional-entropy penalty (7), and $\mathcal{L} = \{cc8, cc9, fc3\}$ as adaptation layers for *multi-layer* MK-MMD/ME penalty. $\mathcal{L}$ is the indices of layers where MMD/ME penalty is active, whose configuration is relying on the size of training data and the number of parameters to be fined-tuned or learned.

ResNet: Deep residual networks [16] have emerged as a family of extremely deep architectures showing compelling accuracy and nice convergence behaviors. The ResNet-50 model contains 50 convolution-pooling layers $conv1$–$pool5$ as feature layers and 1 fully connected layer fc as classifier layer. Again, we opt to add the *multi-layer* MK-MMD/ME penalty on the fully connected layers (fc) and pooling layers (pl) layers. For such an extremely deep adaptation model in Equation (8), we specify the only fully connected layer fc as the deep network classifier f for the cross-entropy loss (5) and the conditional-entropy penalty (7), and $\mathcal{L} = \{pl5, fc\}$ as adaptation layers for *multi-layer* MK-MMD/ME penalty. $\mathcal{L}$ is the indices of layers where MMD/ME penalty is active.

Five crucial advantages clearly distinguish the proposed DAN and VDAN models from the literature. **(1)** *Multi-layer* adaptation. As revealed by [10], feature transferability gets

worse in middle layers ($conv4$–$conv5$ of AlexNet and $in4$–$in6$ of GoogLeNet) and significantly drops in higher layers ($fc6$–$fc8$ of AlexNet and $in7$–$in9$ of GoogLeNet), hence it is critical to adapt multiple layers instead of only one layer. In other words, adapting a single layer cannot remove the dataset shifts between the source and target domains, since there are multiple 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 cross-domain discrepancy underlying *both* the marginal distribution and conditional distribution, which is crucial for domain adaptation [6]. **(2)** *Multi-kernel* two-sample matching. As pointed out by [13], kernel choice is critical to testing effectiveness of MMD/ME since different kernels embed the probability distributions in different RKHSs where different orders of sufficient statistics can be emphasized. This is crucial for moment matching, which has not been explored by previous domain adaptation methods. **(3)** *Adversarial* matching. By jointly learning the distinguishable test locations, our method forms an adversarial learning paradigm where the test locations are learned to maximally discriminate the source and target domains while the deep features are learned to fool the two-sample discrimination. **(4)** *Low-density* separation. As the target-unlabeled data are not identically distributed with the source-labeled data, it is indispensable that the source classifier should pass through the low-density regions of the target domain. Distribution adaptation and low-density separation may reinforce each other for better domain adaptation performance, since they are clearly complementary. **(5)** *Very deep* networks. By going deeper with the convolutions, very deep networks including GoogLeNet [15] and ResNet [16] can learn highly abstract features which are effective not only for class discrimination but also for knowledge transfer. Due to nice trainability of multi-layer MK-MMD/ME penalty and conditional-entropy penalty, it is easy to extend our framework (8) to very deep networks for enhancing feature transferability substantially.

4 Algorithm and Analysis

We present linear-time algorithms for the proposed models based on unbiased estimate of MMD [13], and theoretical analysis on the learning bound for deep domain adaptation.

4.1 Learning Network Parameters

MK-MMD: By the kernel trick $k\left(\mathbf{x},\mathbf{x}'\right)=\left\langle \phi\left(\mathbf{x}\right),\phi\left(\mathbf{x}'\right)\right\rangle$ to avoid explicit manipulation of feature mapping, MK-MMD in Equation (1) can be computed as the expectation of kernel functions, $M_k\left(p,q\right)=\mathbb{E}_{\mathbf{x}^s\mathbf{x}'^s}k(\mathbf{x}^s,\mathbf{x}'^s)+\mathbb{E}_{\mathbf{x}^t\mathbf{x}'^t}k(\mathbf{x}^t,\mathbf{x}'^t)-2\mathbb{E}_{\mathbf{x}^s\mathbf{x}^t}k(\mathbf{x}^s,\mathbf{x}^t)$, where $\mathbf{x}^s,\mathbf{x}'^s\overset{iid}{\sim}p$ and $\mathbf{x}^t,\mathbf{x}'^t\overset{iid}{\sim}q$, $k\in\mathcal{K}$. However, this computation incurs a complexity of $O(n^2)$, which is rather undesirable for deep learning, as the power of deep networks largely derives from learning large-scale datasets. Moreover, the summation over pairwise similarity between all data points makes *mini-batch* stochastic gradient descent (SGD) more difficult, whereas the mini-batch SGD is crucial to the training effectiveness of deep networks. While prior works based on MMD [3], [26], [36] rarely address this problem, we believe it is crucial for deep domain adaptation.

In this paper, we adopt the linear-time unbiased estimate of MK-MMD [13], which can be computed in linear time as

$$\hat{M}_k\left(\mathcal{D}_s,\mathcal{D}_t\right)=\frac{2}{n_s}\sum_{i=1}^{n_s/2}d_k\left(\mathbf{z}_i\right), \tag{9}$$

where we denote quad-tuple $\mathbf{z}_i\triangleq(\mathbf{x}^s_{2i-1},\mathbf{x}^s_{2i},\mathbf{x}^t_{2i-1},\mathbf{x}^t_{2i})$, and evaluate the multi-kernel function k on each quad-tuple $\mathbf{z}_i$ by $d_k\left(\mathbf{z}_i\right)\triangleq k(\mathbf{x}^s_{2i-1},\mathbf{x}^s_{2i})+k(\mathbf{x}^t_{2i-1},\mathbf{x}^t_{2i})-k(\mathbf{x}^s_{2i-1},\mathbf{x}^t_{2i})-k(\mathbf{x}^s_{2i},\mathbf{x}^t_{2i-1})$. This unbiased estimate computes MK-MMD by the expectation of independent variables in (1) with $O(n)$ cost. By computing the linear-time estimate of MK-MMD as the sum of $n_s/2$ quad-tuple functions $d_k(\mathbf{z}_i)$, it fits well into the mini-batch stochastic gradient descent (SGD) procedure.

When training with mini-batch SGD, we only consider the gradient of objective (8) with respect to each point $\mathbf{x}_i$ in a mini-batch $\mathcal{B}\triangleq\{\mathbf{x}_i\}_{i=1}^{|\mathcal{B}|}$ of batch-size $|\mathcal{B}|$. Note that, $\mathcal{B}\subset\mathcal{D}_a\cup\mathcal{D}_u=\mathcal{D}_s\cup\mathcal{D}_t$, that is, each mini-batch may contain both labeled and unlabeled examples from both source and target domains. We maintain a labeled list $\mathcal{B}_a$ and an unlabeled list $\mathcal{B}_u$, $\mathcal{B}=\mathcal{B}_a\cup\mathcal{B}_u$. For each point $\mathbf{x}_i\in\mathcal{B}$, if $\mathbf{x}_i\in\mathcal{B}_a$, then the gradient of empirical error $\frac{\partial L(\mathbf{x}_i,y_i)}{\partial\mathcal{F}^\ell}$ is computed, otherwise $\mathbf{x}_i\in\mathcal{B}_u$, and the gradient of the conditional-entropy penalty $\frac{\partial H(\mathbf{x}_i)}{\partial\mathcal{F}^\ell}$ is computed, both via standard back-propagation.

Since the linear-time MK-MMD (9) can be decoupled as the aggregation of quad-tuple kernel functions $d_k(\mathbf{z}_i)$'s, we merely need to compute the gradients $\frac{\partial d_k(\mathbf{z}_i^\ell)}{\partial\mathcal{F}^\ell}$ for each quad-tuple $\mathbf{z}_i^\ell=\left(\mathbf{h}^{s\ell}_{2i-1},\mathbf{h}^{s\ell}_{2i},\mathbf{h}^{t\ell}_{2i-1},\mathbf{h}^{t\ell}_{2i}\right)$ of the ℓ^{th}-layer hidden representation with respect to the network parameters $\mathcal{F}^\ell=\{\mathbf{W}^\ell,\mathbf{b}^\ell\}$. For each mini-batch $\mathcal{B}$, we traverse through it to sample the quad-tuples $\mathbf{z}_i^\ell$'s until all source points in $\mathcal{B}$ have been accessed. Specifically, we maintain two lists $\mathcal{B}_s$ and $\mathcal{B}_t$ from $\mathcal{B}$, one for the source points and the other for the target points, and $\mathcal{B}=\mathcal{B}_s\cup\mathcal{B}_t$. For each quad-tuple $\mathbf{z}_i^\ell$, we sample two consecutive points $\{\mathbf{h}^{s\ell}_{2i-1},\mathbf{z}^{s\ell}_{2i}\}$ from source list $\mathcal{B}_s$, and two consecutive points $\{\mathbf{h}^{t\ell}_{2i-1},\mathbf{z}^{t\ell}_{2i}\}$ from target list $\mathcal{B}_t$. As we traverse the source list $\mathcal{B}_s$, if we have reached the end of the target list $\mathcal{B}_t$, then we will shuffle the target list and go through it again until the source list has been went through. With this procedure, we can exactly compute the gradients of linear-time MK-MMD (9) using standard mini-batch SGD.

Finally, for each mini-batch $\mathcal{B}$, we combine the gradients of the empirical error (5), entropy penalty (7) and MK-MMD penalty (6) together. For each layer's parameters $\mathcal{F}^\ell$, it yields

$$\begin{aligned}\nabla_{\mathcal{F}^\ell}\left(\mathcal{B}\right)=&\frac{1}{|\mathcal{B}_a|}\sum_{\mathbf{x}_i\in\mathcal{B}_a}1\left\{\mathbf{x}_i\in\mathcal{B}_a\right\}\frac{\partial L\left(\mathbf{x}_i,y_i\right)}{\partial\mathcal{F}^\ell}\\&+\frac{\gamma}{|\mathcal{B}_u|}\sum_{\mathbf{x}_i\in\mathcal{B}_u}1\left\{\mathbf{x}_i\in\mathcal{B}_u\right\}\frac{\partial H\left(\mathbf{x}_i\right)}{\partial\mathcal{F}^\ell}\\&+\frac{2\lambda}{|\mathcal{B}_s|}\sum_{\mathbf{z}_i^\ell\in\mathcal{B}}\frac{\partial d_k\left(\mathbf{z}_i^\ell\right)}{\partial\mathcal{F}^\ell},\end{aligned} \tag{10}$$

where $1\{A\}$ is the indicator function, $1\{A\}=1$ if A is true and $1\{A\}=0$ otherwise. With mini-batch gradient (10), we can perform each mini-batch update by gradient descent for each layer $\mathcal{F}^\ell\leftarrow\mathcal{F}^\ell-\eta\nabla_{\mathcal{F}^\ell}\left(\mathcal{B}\right)$, and η is the learning rate. This can be easily implemented in the Caffe framework for CNNs [23]. Given kernel k as the linear combination of

m Gaussian kernels $\{k_u(\mathbf{x}_i,\mathbf{x}_j)=e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/\sigma_u}\}$, gradient $\frac{\partial d_k(\mathbf{z}_i^\ell)}{\partial \mathcal{F}^\ell}$ can be computed by back-propagation. For example,

$$\begin{aligned}\frac{\partial k(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell})}{\partial \mathbf{W}^\ell} &= \frac{\partial k(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell})}{\partial \tilde{\mathbf{h}}_{2i-1}^{s\ell}}\left(\mathbf{h}_{2i-1}^{s(\ell-1)}\right)^\mathsf{T}\\ &+\frac{\partial k(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell})}{\partial \tilde{\mathbf{h}}_{2i}^{t\ell}}\left(\mathbf{h}_{2i}^{t(\ell-1)}\right)^\mathsf{T},\end{aligned} \tag{11}$$

where the *residual* terms $\frac{\partial k(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell})}{\partial \tilde{\mathbf{h}}_{2i-1}^{s\ell}}$ that back-propagate to influence all the previous layers can be further computed by

$$\begin{aligned}\frac{\partial k(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell})}{\partial \tilde{\mathbf{h}}_{2i-1}^{s\ell}} &= -\sum_{u=1}^{m}\frac{2\beta_u}{\sigma_u}k_u\left(\mathbf{h}_{2i-1}^{s\ell},\mathbf{h}_{2i}^{t\ell}\right)\\ &\times\left[\left(\mathbf{h}_{2i-1}^{s\ell}-\mathbf{h}_{2i}^{t\ell}\right)\odot\dot{a}^\ell\left(\tilde{\mathbf{h}}_{2i-1}^{s\ell}\right)\right],\end{aligned} \tag{12}$$

where $\dot{a}^\ell(\cdot)$ is the derivative of the activation function $a^\ell(\cdot)$, and $\tilde{\mathbf{h}}_i^\ell=\mathbf{W}^\ell\mathbf{h}^{\ell-1}+\mathbf{b}^\ell$ is the layer output before activation.

ME Test: Given J test locations $\mathcal{V}=\{\mathbf{v}_j\}_{j=1}^J\subset\mathbb{R}^J$, the definition of ME test in (4) can be computed in linear time. The computation of gradients of ME with respect to network parameters is similar to that of MK-MMD, which is omitted. The J test locations are learned by maximizing the ME test as $\max_{\mathcal{V}} n\bar{\mathbf{z}}_n^\mathsf{T}\mathbf{S}_n^{-1}\bar{\mathbf{z}}_n$ using gradient ascent, which is done in each iteration or epoch of back-propagation. The gradient of ME-test with respect to each test location $\mathbf{v}_j$ is computed as

$$\begin{aligned}\frac{\partial M_k(\mathcal{D}_s^\ell,\mathcal{D}_t^\ell)}{\partial \mathbf{v}_j} &= -\frac{2}{n}\sum_{i=1}^{n}\left(k(\mathbf{h}_i^{s\ell},\mathbf{v}_j)-k(\mathbf{h}_i^{t\ell},\mathbf{v}_j)\right)\\ &\times\left(\sum_{i=1}^{n}\sum_{u=1}^{m}\frac{2\beta_u}{\sigma_u}k_u(\mathbf{h}_i^{s\ell},\mathbf{v}_j)\times(\mathbf{h}_i^{s\ell}-\mathbf{v}_j)\right.\\ &\left.-\sum_{i=1}^{n}\sum_{u=1}^{m}\frac{2\beta_u}{\sigma_u}k_u(\mathbf{h}_i^{t\ell},\mathbf{v}_j)\times(\mathbf{h}_i^{t\ell}-\mathbf{v}_j)\right).\end{aligned} \tag{13}$$

4.2 Learning Kernel Parameters

Theoretically, the optimal kernel parameter $\boldsymbol{\beta}$ for MK-MMD can be learned by minimizing the Type-II error, i.e. failure of rejecting false null hypothesis $M_k(p,q)\to 0$ when $p\neq q$ [13]. The proposed multi-layer adaptation regularizer performs layerwise distribution matching by minimizing MK-MMD with respect to the network parameter $\mathcal{F}$, while we seek to learn the optimal kernel parameter $\{\boldsymbol{\beta}_\ell\}$ for all the adaptation layers $\ell\in\mathcal{L}$ by minimizing the Type-II error, which is equivalent to maximizing standardized MMD [13],

$$\max_{k\in\mathcal{K}} M_k\left(\mathcal{D}_s^\ell,\mathcal{D}_t^\ell\right)\Sigma_k^{-2}, \tag{14}$$

where $\Sigma_k^2=\mathbb{E}_{\mathbf{z}}d_k(\mathbf{z})-[\mathbb{E}_{\mathbf{z}}d_k(\mathbf{z})]^2$ is estimation covariance. Note that when the covariance is isotropic, i.e. Σ_k equals to the identity matrix, minimizing the Type-II error is reduced to maximizing MK-MMD with respect to kernel parameter $\boldsymbol{\beta}$, which is consistent with Equation (6). For generality, we derive the algorithm based on general covariance structure, which includes (6) as a special case. Denote the m MMDs by $\mathbf{M}=(M_1,M_2,\ldots,M_m)^\mathsf{T}$, and each M_u is MMD for kernel k_u. Covariance matrix $\mathbf{Q}=\mathrm{cov}(M_k)\in\mathbb{R}^{m\times m}$ is computed by $O(m^2n)$, i.e. $\mathbf{Q}_{uu'}=\frac{4}{n_s}\sum_{i=1}^{n_s/4}d_{k_u}^\Delta(\bar{\mathbf{z}}_i)\,d_{k_{u'}}^\Delta(\bar{\mathbf{z}}_i)$, where eight-tuple $\bar{\mathbf{z}}_i\triangleq(\mathbf{z}_{2i-1},\mathbf{z}_{2i})$ and $d_{k_u}^\Delta(\bar{\mathbf{z}}_i)\triangleq d_{k_u}(\mathbf{z}_{2i-1})-d_{k_u}(\mathbf{z}_{2i})$. Problem (14) reduces to a quadratic program (QP),

$$\min_{\mathbf{M}^\mathsf{T}\boldsymbol{\beta}_\ell=1,\boldsymbol{\beta}_\ell\geqslant\mathbf{0}}\boldsymbol{\beta}_\ell^\mathsf{T}(\mathbf{Q}+\varepsilon\mathbf{I})\boldsymbol{\beta}_\ell, \tag{15}$$

where $\varepsilon=10^{-3}$ is a small penalty to make the problem well-defined. The above problem can be efficiently solved by standard QP packages. By solving (15), we obtain a multi-kernel $k=\sum_{u=1}^m\beta_u k_u$ that minimizes the Type-II test error. For ME test, we adopt fix kernel combination parameter $\boldsymbol{\beta}$.

We note that the DAN objective (8) is essentially a minimax problem; i.e., we compute $\min_{f\in\mathcal{F}}\max_{k\in\mathcal{K}} M_k\left(\mathcal{D}_s^\ell,\mathcal{D}_t^\ell\right)\Sigma_k^{-2}$. Since the covariance Σ_k cannot be estimated accurately with typically small-scale datasets in domain adaptation, we find that maximizing (8) and (14) with Σ_k fixed to identity matrix yields more stable results, and will report it in experiments. The CNN parameter $\mathcal{F}$ 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, targeting to consolidate the transferability of DAN features. We accordingly adopt an alternating optimization that updates $\mathcal{F}$ by mini-batch SGD (10) and $\boldsymbol{\beta}$ by QP (15) iteratively. Both updates cost $O(n)$ and are scalable to large-scale data.

4.3 Generalization Error Analysis

We provide a brief analysis on target risk bound, by making use of the theory of domain adaptation [11], [12], [2] and the theory of kernel embedding of distributions [40], [13], [38].

Theorem 1. *[11], [12] Let $f\in\mathcal{H}$ be a hypothesis, $R_s(f)$ and $R_t(f)$ be the expected risks of source and target respectively, then*

$$R_t(f)\leqslant R_s(f)+d_{\mathcal{H}}(p,q)+C, \tag{16}$$

where C is the expected risk of ideal hypothesis for both domains, and $d_{\mathcal{H}}(p,q)$ is the $\mathcal{H}$-divergence between distributions p and q,

$$d_{\mathcal{H}}(p,q)\triangleq 2\sup_{\eta\in\mathcal{H}}\left|\Pr_{\mathbf{x}^s\sim p}[\eta(\mathbf{x}^s)=1]-\Pr_{\mathbf{x}^t\sim q}[\eta(\mathbf{x}^t)=1]\right|. \tag{17}$$

Hypothesis $\eta\in\mathcal{H}$ is essentially a *two-sample* classifier that discriminates the source and target. The $\mathcal{H}$-divergence largely depends on the capacity of the hypothesis space $\mathcal{H}$ to characterize the discrepancy between distributions p and q. The hypothesis space $\mathcal{H}$ should be rich enough so that any key difference underlying p and q can be captured. We choose (kernel) Parzen window classifier as the two-sample classifier η [40], which is *nonparametric* and is rich enough to capture the distribution difference. We show that $d_{\mathcal{H}}(p,q)$ can be bounded by the empirical error of the (kernel) Parzen window classifier, which is equivalent to the MK-MMD [40]:

$$\begin{aligned}d_{\mathcal{H}}(p,q) &\leqslant \hat{d}_{\mathcal{H}}(\mathcal{D}_s,\mathcal{D}_t)+C_0\\ &\leqslant 2-2\inf_{\eta\in\mathcal{H}}\left(\sum_{i=1}^{n_s}\frac{L[\eta(\mathbf{x}_i^s)=1]}{n_s}+\sum_{j=1}^{n_t}\frac{L[\eta(\mathbf{x}_j^t)=-1]}{n_t}\right)+C_0\\ &= 2\left(1+M_k(\mathcal{D}_s,\mathcal{D}_t)\right)+C_0,\end{aligned} \tag{18}$$

where C_0 is a complexity term that bounds $\mathcal{H}$-divergence from its empirical estimate, $L(\cdot)$ is the linear loss function of the Parzen window classifier η, where $L[\eta=1]\triangleq-\eta$, $L[\eta=-1]\triangleq\eta$. By learning deep features that explicitly minimize MK-MMD in multiple layers, the deep features learned by the DAN and VDAN models can decrease the upper bound of the target-domain risk. Note that we maximize MK-MMD w.r.t. $\boldsymbol{\beta}$ in Equation (14), which helps the Parzen window classifier achieve minimal risk of two-sample discrimination

in (18). An important advantage by choosing *nonparametric* (kernel) Parzen window as the two-sample classifier is that it is almost parameter-free (except a few kernel-combination parameters $\boldsymbol{\beta}$) and no new network parameters are introduced, which makes the DAN and VDAN models easy to train. In contrast, domain-adversarial network [37] chooses a three-layer perceptron as the two-sample discriminator, which has more parameters and is not easy to train as [30].

5 Experiments

We evaluate the proposed deep adaptation networks against state of the art transfer learning and deep learning methods on both unsupervised and semi-supervised domain adaptation tasks. Datasets, codes, and configurations will be made available at http://ise.thss.tsinghua.edu.cn/~mlong/vdan.

5.1 Datasets

We adopt three benchmark datasets: *Office-31*, *Office-Caltech* and *ImageCLEF-DA*. For all deep transfer methods, we use original images as input. For all shallow transfer methods, we use deep features by AlexNet [22] and GoogLeNet [15].

Office-31 [27] is an open benchmark for domain adaptation, which comprises 4,652 images over 31 classes collected from three domains: *Amazon* (**A**), which contains images downloaded from amazon.com, *Webcam* (**W**) and *DSLR* (**D**), which contain images respectively taken by a web camera and a digital SLR camera in some office environment. We evaluate all methods across three transfer tasks **A** $\rightarrow$ **W**, **D** $\rightarrow$ **W** and **W** $\rightarrow$ **D**, which are commonly adopted by deep learning methods [9], [36], [37], and across the other three transfer tasks **A** $\rightarrow$ **D**, **D** $\rightarrow$ **A** and **W** $\rightarrow$ **A** as studied in [30].

Office-Caltech [29], a dataset widely adopted by domain adaptation methods [43], [32], is built by selecting the 10 common categories shared by *Office-31* and *Caltech-256* (**C**) [44]. We consider all domain combinations of this dataset, and construct 12 transfer tasks: **A** $\rightarrow$ **W**, **D** $\rightarrow$ **W**, **W** $\rightarrow$ **D**, **A** $\rightarrow$ **D**, **D** $\rightarrow$ **A**, **W** $\rightarrow$ **A**, **A** $\rightarrow$ **C**, **W** $\rightarrow$ **C**, **D** $\rightarrow$ **C**, **C** $\rightarrow$ **A**, **C** $\rightarrow$ **W**, and **C** $\rightarrow$ **D**. While *Office-31* has more categories and is more difficult for domain adaptation, *Office-Caltech* provides more transfer tasks for an unbiased look at dataset bias [45].

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 four public datasets, each is considered as a domain: *Caltech-256* (**C**), *ImageNet ILSVRC 2012* (**I**), *Pascal VOC 2012* (**P**), and *Bing* (**B**). We consider all domain combinations and build 12 transfer tasks: **C** $\rightarrow$ **I**, **C** $\rightarrow$ **P**, **C** $\rightarrow$ **B**, **I** $\rightarrow$ **C**, **I** $\rightarrow$ **P**, **I** $\rightarrow$ **B**, **P** $\rightarrow$ **C**, **P** $\rightarrow$ **I**, **P** $\rightarrow$ **B**, **B** $\rightarrow$ **C**, **B** $\rightarrow$ **I**, and **B** $\rightarrow$ **P**. It is worth noting that, some domains (e.g. **B**) are low-quality images that are more difficult to categorize than other domains (e.g. **C**). This makes it a good complement to the *Office-31* dataset.

5.2 Experimental Setup

5.2.1 Comparison Methods

We compare with state of the art transfer learning and deep learning methods: Transfer Component Analysis (**TCA**) [3], Geodesic Flow Kernel (**GFK**) [29], Subspace Alignment (**SA**) [46], Deep Domain Confusion (**DDC**) [36], Reverse Gradient (**RevGrad**) [37]. TCA learns a shared feature space by MMD-penalized Kernel PCA. GFK interpolates across an infinite number of intermediate subspaces to bridge the source and target subspaces. SA seeks a domain invariant feature space by aligning the source subspace with the target subspace. For these three methods, we adopt SVM as base classifier. DDC maximizes domain confusion by adding to AlexNet an adaptation layer that is regularized by linear-kernel MMD. RevGrad enables domain adversarial learning by adapting a single layer of AlexNet, which matches source and target by making them indistinguishable for a domain discriminator.

We examine the influence of deep representations for domain adaptation, by studying **AlexNet** [22], **GoogLeNet** [15], and **ResNet** [16] as base architectures for learning deep representations. For shallow methods, we follow DeCAF [9] and use as deep image representations the activations of the $fc7$ (AlexNet), $in9$ (GoogLeNet), and $pool5$ (ResNet) layers.

We go deeper into our models by studying their variants. To testify *multi-layer* adaptation, we run DAN and VDAN using only one adaptation layer, $fc7$ of AlexNet or $in9$ of GoogLeNet, respectively termed **DAN-fc7** and **VDAN-in9**. To testify *multi-kernel* MMD, we evaluate DAN and VDAN using single-kernel MMD, respectively termed **DAN-sk** and **VDAN-sk**. To testify *entropy minimization*, we evaluate DAN and VDAN by removing the entropy penalty, respectively termed **DAN-ent** and **VDAN-ent**. Our preliminary paper [24] is DAN-ent that does not use entropy penalty and very deep networks. All above variants are based on MMD, while the one replacing MMD with ME is termed as **VDAN (ME)**.

5.2.2 Evaluation Protocols

We follow standard evaluation protocols [5], [27]. In the *non-sampling* protocol [5], we utilize all labeled source examples and all unlabeled target examples for unsupervised domain adaptation; and for semi-supervised domain adaptation, we further sample 3 labeled examples per category from target domain [27]. In the *down-sampling* protocol [27] for *Office-31*, we randomly sample as the source domain 20 labeled examples per category from Amazon (**A**) and 8 labeled examples per category from Webcam (**W**) and DSLR (**D**). We compare the average classification accuracy and the standard deviation of each method on all random experiments.

For all baseline methods, we follow their original model selection procedures, or conduct *transfer cross-validation* [47] if their model selection strategies are not specified. We also adopt *transfer cross-validation* [47] to select parameters λ and γ for the DAN and VDAN models. Fortunately, our models perform very stably under different parameters, thus we fix $\lambda = 1$, $\gamma = 0.1$ throughout all experiments. For MMD-based methods (TCA, DDC, DAN, and VDAN), we use the Gaussian (RBF) kernel $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma}$ with the bandwidth σ set to median pairwise squared distances on training data, i.e. the *median heuristic* [13]. We use multi-kernel MMD and multi-kernel ME for DAN and VDAN, and a family of m Gaussian kernels $\{k_u\}_{u=1}^{m}$ varying bandwidth $\sigma_u \in [2^{-8}\sigma, 2^{8}\sigma]$ with a multiplicative step-size of $2^{1/2}$ [13].

We implement all deep methods based on the **Caffe** [23] deep-learning framework, and fine-tune from AlexNet [22], GoogLeNet [15] and ResNet [16] models pre-trained on the ImageNet dataset [18]. We fine-tune all convolutional

1. http://imageclef.org/2014/adaptation

TABLE 1
Accuracy on *Office-31* Dataset under **Non-Sampling** Protocol [5] for Unsupervised and Semi-Supervised Domain Adaptation (**AlexNet**)

Method	Unsupervised Domain Adaptation							Semi-Supervised Domain Adaptation						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet [22]	60.6±0.5	95.4±0.3	99.0±0.3	64.2±0.4	45.5±0.6	48.3±0.5	68.8	81.6±0.3	97.0±0.2	99.6±0.2	80.7±0.3	64.3±0.4	64.1±0.3	81.2
GFK [29]	58.4±0.0	93.6±0.0	91.0±0.0	58.6±0.0	**52.4**±0.0	46.1±0.0	66.7	81.0±0.3	94.2±0.2	96.3±0.4	81.0±0.5	65.5±0.4	64.5±0.6	80.4
SA [46]	58.5±0.0	92.0±0.0	98.8±0.0	60.1±0.0	50.1±0.0	47.3±0.0	67.9	80.2±0.6	95.8±0.5	98.8±0.4	78.9±0.3	65.2±0.4	63.3±0.4	80.4
TCA [3]	59.0±0.0	90.2±0.0	88.2±0.0	57.8±0.0	51.6±0.0	47.9±0.0	65.8	82.5±0.3	94.9±0.2	97.3±0.3	80.4±0.4	65.8±0.4	65.1±0.5	81.0
DDC [36]	61.0±0.5	95.0±0.3	98.5±0.3	64.9±0.4	47.2±0.5	49.4±0.6	69.3	85.9±0.4	96.5±0.3	99.2±0.2	80.8±0.4	64.3±0.4	64.5±0.4	81.9
RevGrad [37]	73.0±0.5	96.4±0.3	98.5±0.3	-	-	-	-	-	-	-	-	-	-	-
DAN-fc7	70.2±0.4	96.5±0.2	**99.6**±0.2	71.3±0.3	47.4±0.3	51.3±0.4	72.7	88.4±0.3	97.0±0.1	99.9±0.1	83.5±0.2	65.9±0.3	67.8±0.3	83.8
DAN-sk	70.4±0.7	96.7±0.4	99.6±0.4	70.1±0.5	48.1±0.6	50.2±0.6	72.5	89.5±0.3	97.4±0.2	**100.0**±0.0	83.7±0.4	66.5±0.5	**68.3**±0.4	84.2
DAN-ent [24]	68.5±0.5	96.0±0.3	99.0±0.3	66.8±0.4	50.0±0.5	49.8±0.5	71.7	86.3±0.4	97.2±0.2	99.6±0.2	82.1±0.4	64.6±0.3	65.2±0.3	82.5
DAN	**73.9**±0.5	**96.8**±0.3	**99.6**±0.2	**71.7**±0.4	50.0±0.6	**51.4**±0.6	**73.9**	**89.7**±0.4	**97.5**±0.2	**100.0**±0.0	**84.1**±0.3	**67.8**±0.4	**68.3**±0.4	**84.6**

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 lower layers. We employ the mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate strategy implemented in RevGrad [37]: 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 parameters λ and γ, we gradually change them from 0 to $\lambda = 1, \gamma = 0.1$ by multiplying $\frac{2}{1+\exp(-\delta p)} - 1$, where $\delta = 10$ [37]. This new parameter strategy significantly stabilizes the parameter sensitivity of the proposed models.

5.3 Results and Discussion

We elaborate unsupervised and semi-supervised adaptation results on *Office-31* dataset, and discuss unsupervised adaptation results on *Office-Caltech* and *ImageCLEF-DA* datasets.

5.3.1 Results on Office-31 with AlexNet

As AlexNet [22] architecture can learn generalizable features [10], we compare all methods using AlexNet features as input (i.e. DeCAF features [9]) or using AlexNet as base architecture. Table 1 details the classification accuracy results of both unsupervised domain adaptation and semi-supervised domain adaptation under the non-sampling protocol, where the results of RevGrad are directly reported from its original paper [37]. The DAN model based on AlexNet (i.e. Figure 1) outperforms all comparison methods on most transfer tasks, and significantly improves classification accuracy on 3 out of 6 transfer tasks. For unsupervised domain adaptation, DAN achieves an absolute accuracy increase of **4.6%** against the best baseline DDC. For semi-supervised domain adaptation, DAN outperforms the best baseline DDC by an absolute accuracy lift of **2.7%**. This validates that DAN can learn more transferable features for more effective domain adaptation.

From the experimental results, we can make insightful observations. **(1)** Traditional transfer-learning methods with deep AlexNet features as input may either underperform or outperform standard deep-learning methods. Traditional transfer-learning methods may benefit from domain adaptation procedure while standard deep-learning methods may take advantage of fine-tuning. It shows that the two worlds (deep and transfer) cannot reinforce each other in the two-stage pipeline, and require end-to-end architectures to unify the two worlds. **(2)** Deep-transfer learning methods that reduce the domain discrepancy by domain-adaptive deep networks (DDC and RevGrad) generally outperform traditional transfer-learning methods with AlexNet features as input. This confirms that incorporating domain-adaptation modules to deep networks can improve feature transferability. However, different layers of deep networks extract features at different abstraction levels, and thus the extracted features may not be safely transferable in multiple layers. A natural and safe approach is to reduce the cross-domain discrepancy at each task-specific layer to consolidate transfer. **(3)** By adapting the cross-domain distributions in multiple task-specific layers using optimal multi-kernel two-sample matching and exploiting the cluster structure (low-density separation) of the target-unlabeled data, the proposed DAN model establishes new state of art on the *Office-31* dataset.

To go deeper into the modules of DAN, we demonstrate the results of DAN variants: DAN-fc7 (single-layer), DAN-sk (single-kernel), and DAN-ent (without entropy penalty). The results in Table 1 well justify our motivation. **(1)** DAN-fc7 achieves much better results than DDC, but substantially underperforms DAN. This validates the importance of *multi-layer* adaptation of all task-specific layers for learning transferable features. Deep networks can perform moment-matching (match all statistics of distributions) based on all the abstraction levels of the source and target data. **(2)** DAN-sk achieves much better accuracies than DDC, but substantially underperforms DAN. This testifies that *multi-kernel* distribution matching can reduce cross-domain discrepancy more effectively than single-kernel distribution matching. Distribution matching by multiple kernels of different bandwidths match all low-order and high-order moments. The evidence that DAN-sk outperforms DAN-fc7 justifies that multi-layer adaptation brings more performance gains than multi-kernel distribution matching. **(3)** DAN-ent achieves much higher accuracies than DDC, but significantly underperforms DAN. This highlights the importance of low-density separation by entropy minimization, which exploits the cluster structure of target-unlabeled data such that the source-classifier can be better adapted to the target-data. **(4)** The full DAN model achieves the best results by jointly performing multi-layer adaptation with multi-kernel MMD and exploiting low-density separation of target-unlabeled data. By jointly adapting the representation layers ($conv1$–$fc7$) and the classifier layer ($fc8$), we essentially reduce the cross-domain discrepancy underlying *both* marginal distribution (of features) and conditional distribution (of labels given features), which is key to domain adaptation [6], [7].

TABLE 2
Accuracy on *Office-31* Dataset under **Down-Sampling** [27] and **Non-Sampling** [5] Protocols for Unsupervised Domain Adaptation (**GoogLeNet**)

Method	Down-Sampling Protocol							Non-Sampling Protocol						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	A → W	D → W	W → D	A → D	D → A	W → A	Avg
GoogLeNet [15]	70.1±0.9	**95.1**±0.5	**98.2**±0.3	71.1±0.5	59.9±0.5	59.4±0.6	75.6	71.4±0.5	95.8±0.2	98.3±0.2	72.2±0.3	61.0±0.3	60.5±0.3	76.5
GFK [29]	63.4±0.6	90.1±0.4	92.6±0.5	65.9±0.6	59.0±0.6	56.8±0.5	71.3	71.2±0.0	96.4±0.0	**99.0**±0.0	70.3±0.0	62.7±0.0	60.7±0.0	76.7
TCA [3]	63.5±0.4	87.1±0.2	92.3±0.4	66.2±0.4	59.2±0.5	58.2±0.5	71.1	68.6±0.0	94.0±0.0	97.4±0.0	69.5±0.0	61.7±0.0	61.4±0.0	75.4
DDC [36]	70.0±0.7	93.8±0.5	96.2±0.4	71.0±0.5	62.8±0.6	61.9±0.6	76.0	72.5±0.4	95.5±0.2	98.1±0.1	73.2±0.3	61.6±0.3	61.6±0.3	77.1
VDAN-in9	71.8±0.6	94.0±0.4	96.4±0.3	72.2±0.5	64.2±0.6	63.8±0.6	77.0	74.8±0.3	96.0±0.2	98.5±0.1	74.7±0.2	62.4±0.3	63.8±0.3	78.4
VDAN-sk	72.2±0.6	93.5±0.4	97.6±0.3	72.9±0.5	**65.7**±0.6	61.4±0.3	77.2	72.6±0.2	95.8±0.2	99.4±0.1	74.9±0.2	65.9±0.2	63.0±0.3	78.6
VDAN-ent	73.2±0.5	94.3±0.4	97.2±0.3	72.9±0.5	64.7±0.5	62.3±0.6	77.4	76.0±0.3	95.9±0.2	98.6±0.1	74.4±0.2	61.5±0.3	60.3±0.2	77.8
VDAN	**73.6**±0.7	94.5±0.4	96.6±0.4	**73.5**±0.6	65.4±0.7	**64.4**±0.6	**78.0**	**80.6**±0.3	**96.2**±0.2	98.6±0.1	**75.8**±0.3	**66.2**±0.2	**66.5**±0.3	**80.6**

Theoretical analysis of domain adaptation [2], [12] shows that cross-domain discrepancy plays a key role in bounding the transfer error. The *Office-31* dataset has considered the cross-domain discrepancy. By construction, domains **W** and **D** are very similar to each other, while they are significantly dissimilar to domain **A**. The results in Table 1 well testify the above theory: when the cross-domain discrepancy is small (**W** → **D** and **D** → **W**), the transfer performance is excellent; when the cross-domain discrepancy is large (**A** → **W** and **A** → **D**), the transfer performance drops remarkably. It is very desirable that the DAN model improves the transfer performance more significantly for the transfer tasks with larger cross-domain discrepancy (e.g. task **A** → **W**). Another interesting observation is the *asymmetric* property of domain adaptation: the difficulty of transferring from domain **S** to **T** differs from that of **T** to **S**. Transferring from large domain to small domain is easier, e.g. **A** → **W** is easier than **W** → **A**. This is because although the cross-domain discrepancy between a source-target pair is identical, the risk of a large source domain is lower and results in lower target risk [12].

In many applications, a few target-labeled examples may be available. We further report the semi-supervised domain adaptation results under the non-sampling protocol [27] in Table 1. The results reveal interesting observations. **(1)** The methods under semi-supervised domain adaptation outperform the counterparts under unsupervised domain adaptation by huge margins. This implies that the target-labeled examples are much more "valuable" than the source-labeled examples. **(2)** But we should not depend only on the target-labeled examples if they are very scarce as limited target supervision cannot induce reliable target classifier without overfitting. **(3)** The rich source-labeled examples give a low-variance but domain-biased prior for the target classifier. The domain bias is further corrected by DAN such that the source classifier can be better adapted to the target domain. **(4)** The target-labeled examples specify where an accurate target-classifier may reside in the source-induced hypothesis space, which boosts the transfer performance significantly.

5.3.2 Results on Office-31 with GoogLeNet

Though the transferability of AlexNet features has been well quantified [10], it remains unclear whether GoogLeNet [15] can learn more transferable features than AlexNet and how feature transferability may change with very deep networks. In this paper, we approach this goal by examining all comparison methods using GoogLeNet features as input (similar to the off-the-shelf DeCAF features [9]) or using GoogLeNet as their building architecture (DDC and VDAN). For DDC [36], a state of the art method for comparison, although its original paper did not implement it based on GoogLeNet, its architecture is similar to DAN and we are able to reproduce it based on GoogLeNet and generate reasonable new results.

Table 2 reports unsupervised domain adaptation results on the *Office-31* dataset under both down-sampling [27] and non-sampling [5] protocols. For down-sampling protocol, VDAN outperforms the best baseline DDC by **2.0%**. For non-sampling protocol, VDAN outperforms the best baseline DDC by **3.5%**. Compared with the results of AlexNet-based methods in Table 1, we can make several interesting observations. **(1)** GoogLeNet-based methods outperform AlexNet-based methods by very large margins (∼**6%**). This shows that *very deep* convolutional networks (GoogLeNet [15] and VGG [48]) not only learn better representations for general computer vision tasks but also learn more transferable representations for effective domain adaptation. **(2)** The VDAN model significantly outperforms GoogLeNet-based baseline methods, which reveals that even very deep networks can only reduce, but not remove, the domain discrepancy. **(3)** The boost of VDAN over GoogLeNet is less significant than the improvement of DAN over AlexNet. This implies that GoogLeNet can learn more transferable features than AlexNet. **(4)** The behaviors of different VDAN variants conform with that of the DAN variants. This implies that the efficacy of different adaptation criteria (i.e. multi-layer deep adaptation, multi-kernel two-sample matching, and entropy minimization) can generalize to various deep architectures.

It is interesting to study how the transfer performance will change with respect to the size of source-labeled examples, which can reflect the "market value" of source dataset [45], [29]. To this end, Table 2 compares the results of unsupervised domain adaptation using both down-sampling [27] and non-sampling protocols [5]. The average accuracy under the non-sampling protocol is **80.6%**, which is merely **2.6%** higher than that under the down-sampling protocol [27]. This implies the "marginal utility" of the source-labeled examples saturates quickly and adding more source-labeled examples can hardly improve the transfer performance. It is worth noting that, by adding only 3 target-labeled examples, the transfer accuracy is dramatically boosted by ∼**11%** (Table 1). This reveals an intrinsic limitation of unsupervised domain adaptation for practical applications, and calls for semi-supervised domain adaptation for better performance.

5.3.3 Results on Office-31 with ResNet

ResNet [16] is the top-performing architecture that achieves new state of the art for classification and detection in ImageNet Large-Scale Visual Recognition Challenge 2015 [18]. Although it has been shown that very deep networks can learn more transferable features for domain adaptation, it remains unclear whether extremely deep networks such as

TABLE 3
Accuracy on *Office-31* Dataset under **Non-Sampling** [5] Protocol for Unsupervised Domain Adaptation (**ResNet**)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet [16]	68.4±0.3	96.7±0.2	99.3±0.1	68.9±0.3	62.5±0.4	60.7±0.4	76.1
RevGrad [37]	82.0±0.5	96.9±0.4	99.1±0.3	79.7±0.2	**68.2**±0.6	67.4±0.5	82.2
VDAN (MMD)	86.3±0.3	**97.2**±0.2	99.6±0.1	**82.1**±0.3	64.6±0.4	65.2±0.3	82.5
VDAN (ME)	**89.2**±0.3	96.6±0.2	**100.0**±0.0	82.0±0.3	66.1±0.4	**67.6**±0.4	**83.6**

TABLE 4
Classification Accuracy on *Office-Caltech* Dataset under **Non-Sampling** Protocol [5] for Unsupervised Domain Adaptation (**AlexNet**)

Method	A → W	D → W	W → D	A → D	D → A	W → A	A → C	W → C	D → C	C → A	C → W	C → D	Avg
AlexNet [22]	83.1±0.2	97.7±0.2	**100.0**±0.0	88.5±0.3	89.3±0.2	83.8±0.3	84.6±0.2	77.7±0.3	80.9±0.3	91.8±0.2	83.1±0.2	89.0±0.2	87.5
GFK [29]	89.5±0.0	97.0±0.0	98.1±0.0	86.0±0.0	89.8±0.0	88.5±0.0	76.2±0.0	77.1±0.0	77.9±0.0	90.7±0.0	78.0±0.0	77.1±0.0	85.5
SA [46]	81.7±0.0	97.0±0.0	**100.0**±0.0	87.3±0.0	84.6±0.0	82.1±0.0	82.1±0.0	74.1±0.0	76.6±0.0	92.1±0.0	84.1±0.0	86.6±0.0	85.7
TCA [3]	84.4±0.0	96.9±0.0	99.4±0.0	82.8±0.0	90.4±0.0	85.6±0.0	81.2±0.0	75.5±0.0	79.6±0.0	92.1±0.0	88.1±0.0	87.9±0.0	87.0
DDC [36]	86.1±0.3	98.2±0.1	**100.0**±0.0	89.0±0.2	89.5±0.2	84.9±0.3	85.0±0.2	78.0±0.3	81.1±0.3	91.9±0.2	85.4±0.2	88.8±0.3	88.2
DAN-fc7	92.2±0.2	98.9±0.1	**100.0**±0.0	90.3±0.2	92.6±0.1	88.8±0.2	86.3±0.3	83.7±0.3	82.3±0.2	93.1±0.2	94.3±0.1	90.5±0.2	91.1
DAN-sk	94.2±0.2	98.9±0.1	**100.0**±0.0	91.7±0.2	94.0±0.1	92.7±0.2	86.2±0.3	83.0±0.3	81.9±0.3	92.4±0.2	95.0±0.2	90.5±0.2	91.7
DAN-ent [24]	93.8±0.2	**99.0**±0.1	**100.0**±0.0	92.4±0.2	92.0±0.2	92.1±0.2	85.1±0.3	84.3±0.3	**82.4**±0.4	92.0±0.2	90.6±0.2	90.5±0.2	91.2
DAN	**96.1**±0.1	**99.0**±0.1	**100.0**±0.0	**92.8**±0.2	**94.3**±0.2	**93.4**±0.2	88.0±0.3	**87.3**±0.3	**82.4**±0.3	**93.5**±0.2	**96.3**±0.1	**91.4**±0.3	**92.9**

TABLE 5
Classification Accuracy on *ImageCLEF-DA* Dataset under **Non-Sampling** Protocol [5] for Unsupervised Domain Adaptation (**GoogLeNet**)

Method	C → I	C → P	C → B	I → C	I → P	I → B	P → C	P → I	P → B	B → C	B → I	B → P	Avg
GoogLeNet [15]	85.2±0.2	66.8±0.3	58.6±0.3	92.2±0.2	76.5±0.3	57.2±0.3	91.3±0.2	85.0±0.3	**58.7**±0.3	87.0±0.2	81.3±0.3	65.2±0.3	75.4
GFK [29]	84.2±0.0	69.8±0.0	58.3±0.0	92.5±0.0	75.8±0.0	57.3±0.0	82.2±0.0	78.7±0.0	48.8±0.0	81.2±0.0	73.5±0.0	61.5±0.0	72.0
TCA [3]	83.2±0.0	69.8±0.0	58.8±0.0	93.7±0.0	76.0±0.0	60.2±0.0	93.9±0.0	84.0±0.0	56.8±0.0	87.0±0.0	80.3±0.0	67.0±0.0	75.9
DDC [36]	86.3±0.2	71.5±0.3	58.9±0.3	92.8±0.2	77.7±0.3	59.6±0.3	89.9±0.3	83.7±0.2	55.5±0.3	87.6±0.2	80.4±0.2	69.1±0.3	76.1
VDAN-in9	90.5±0.2	**73.6**±0.3	**62.4**±0.2	**97.0**±0.2	78.5±0.3	60.8±0.3	92.6±0.2	86.2±0.2	56.6±0.3	91.2±0.2	84.0±0.2	73.5±0.2	78.9
VDAN-sk	89.5±0.2	71.8±0.2	60.5±0.2	96.0±0.2	76.4±0.3	51.2±0.3	**95.3**±0.2	**89.3**±0.2	55.4±0.3	92.6±0.2	**85.6**±0.2	72.7±0.2	78.0
VDAN-ent	90.0±0.1	73.5±0.3	60.8±0.3	96.1±0.1	78.9±0.2	58.6±0.2	92.7±0.2	84.7±0.3	56.3±0.2	90.0±0.3	82.4±0.2	**73.7**±0.3	78.1
VDAN	**91.2**±0.2	72.3±0.2	61.3±0.2	96.6±0.1	**79.3**±0.2	**61.5**±0.3	95.0±0.1	89.2±0.2	58.3±0.2	**93.8**±0.2	85.2±0.3	73.3±0.2	**79.7**

ResNet (50 layers) can completely remove the dataset bias across different domains. We thus evaluate the most competing method RevGrad using ResNet as building architecture, and omit the other baseline methods for space limitation. We also investigate the new ME discrepancy in this experiment.

Table 3 reports unsupervised domain adaptation results on the *Office-31* dataset under the non-sampling [5] protocol. RevGrad and VDAN significantly outperform the standard ResNet by very large margins (∼**6%**). This reveals that even extracting highly abstract features of the source and target domains using extremely deep networks, the cross-domain discrepancy still lingers in deep features. This validates that domain adaptation itself is a very challenging problem that cannot be solved by deep learning alone. Although VDAN based on MMD performs comparably with RevGrad, VDAN based on the new ME discrepancy significantly outperforms RevGrad, especially on task **A** → **W**. The ME discrepancy can learn interpretable features so as to maximize the distinguishability of source and target distributions. Adapting domains against the features where their distributions maximally differ, the learned features will be most transferable.

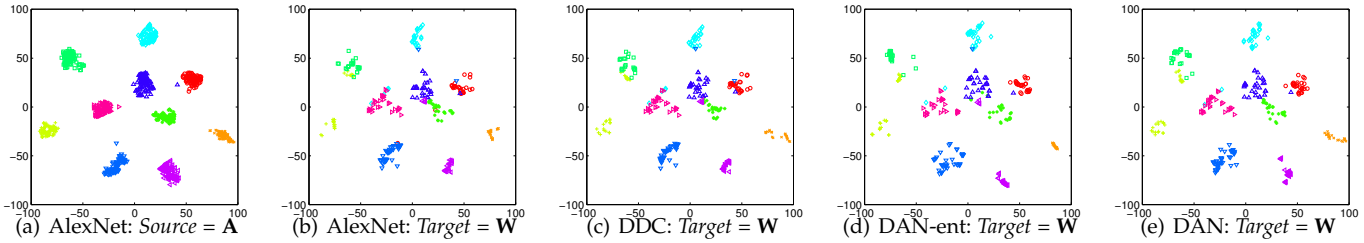
5.3.4 Results on Office-Caltech with AlexNet

Since these tasks are easier than *Office-31*, we only report the results based on AlexNet. Table 4 lists the results of unsupervised domain adaptation using non-sampling protocol [5]. The DAN model outperforms comparison methods on most transfer tasks, and substantially improves the classification accuracy on 8 out of 12 transfer tasks. The average classification accuracy of DAN on all 12 transfer tasks is **92.9%**, and the absolute accuracy increase is **4.7%** against the best baseline DDC. This validates that the DAN model can learn transferable features to boost domain adaptation. By comparing Tables 1–4, we find that traditional transfer-learning methods with deep features as the input cannot consistently outperform standard deep-learning methods. Unlike end-to-end methods, traditional transfer-learning methods may be sensitive to different architectures and different datasets.

5.3.5 Results on ImageCLEF-DA with GoogLeNet

The four domains in *ImageCLEF-DA* are more balanced than *Office-Caltech* but some domains (e.g. **B**) comprise low-quality images that are much more difficult to categorize. With more difficult transfer tasks, we are expecting to testify whether transfer learning helps for harder problems. As the transfer tasks are more difficult, we report the results with GoogLeNet. Table 5 compares the results of unsupervised domain adaptation using the non-sampling protocol [5]. The VDAN model outperforms all comparison methods on most transfer tasks, and substantially improves the classification accuracy on 7 out of the 12 transfer tasks. The average classification accuracy of VDAN on all 12 transfer tasks is **79.7%**, attaining an absolute accuracy improvement of **3.6%** against the best baseline DDC. This new evidence on *ImageCLEF-DA* validates that the VDAN model can learn transferable features for effective domain adaptation on hard problems.

An interesting result that we reproduce is the *asymmetric* property of domain adaptation, but in a different paradigm: the difficulty degree of transferring from easy domain **S** to hard domain **T** is significantly lower than transferring from **T** to **S**. For example, based on the quality of images and their labeling, the degrees of difficulty for the four domains are **B** > **P** > **I** > **C**. Hence, when **B** is used as the target domain, the classification accuracy is much lower than that of when **B** is used as the source domain. Although the cross-domain discrepancy (MMD) between a particular source-target pair



(a) AlexNet: *Source* = **A** (b) AlexNet: *Target* = **W** (c) DDC: *Target* = **W** (d) DAN-ent: *Target* = **W** (e) DAN: *Target* = **W**

Fig. 3. Visualization analysis: (a) t-SNE of source features by AlexNet; (b)–(e) t-SNE of target features by AlexNet, DDC, DAN-ent, DAN, respectively.

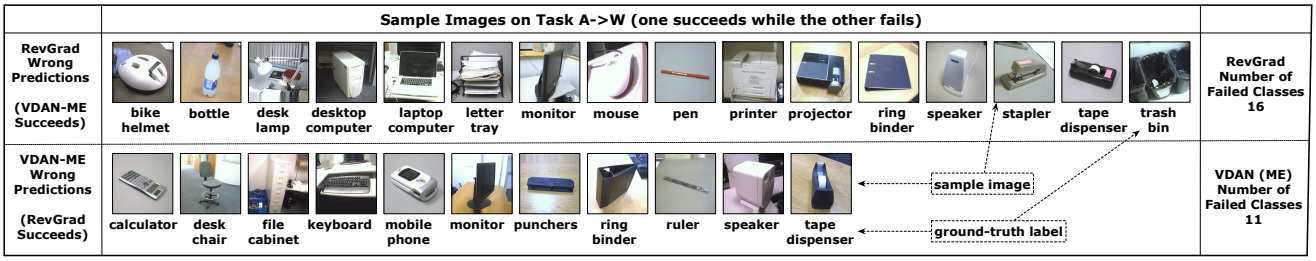


Fig. 4. Sample images where our method succeeds while the baseline method fails or vice versa, for an in-depth look into the error-prone samples.

is identical while the expected risk of an easy source domain may be lower, the expected risk of a hard target domain may still be higher, due to the irreducible term in the domain-adaptation learning bound that quantifies the adaptability [2], [12]. This first implies that domain adaptation is more effective when the target domain is "easier" than the source domain, and points out an open problem of how to address the asymmetric case by domain-adaptation learning theory.

5.4 Empirical Analysis

5.4.1 Visualization Analysis

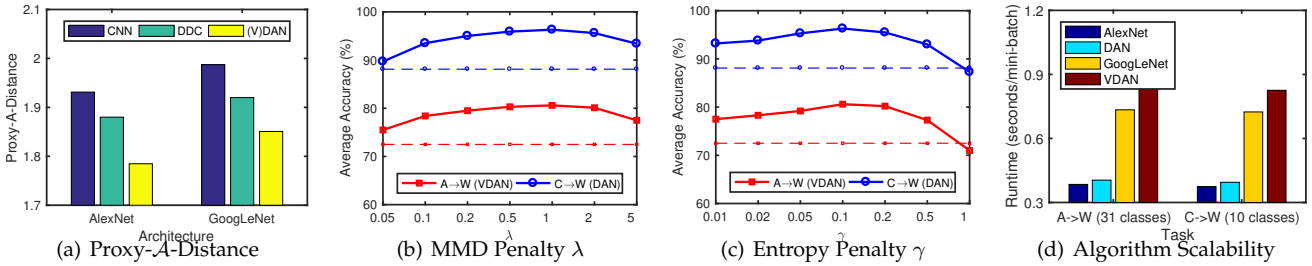
We demonstrate the feature transferability by visualizing in Figures 3(a)–3(e) the t-SNE embeddings [9] of the images in transfer task $\mathbf{A} \rightarrow \mathbf{W}$ with the deep features by AlexNet (**A** & **W**), DDC (**W**), DAN-ent (**W**), and DAN (**W**), respectively. We can make the following observations. (1) With AlexNet features in Figure 3(a), the target categories are not well aligned with the source categories, which implies that target data is not compatible with the source classifier. (2) DDC improves AlexNet by matching only the means of the source and target data under the AlexNet-fc7 features. However, two problems remain, which can be seen from Figure 3(c): The power of matching only the means of the source and target distributions is relatively weak [38]; Matching only one layer of the deep network is also not safe since there may be multiple layers where the features are not transferable [10]. (3) DAN-ent further improves DDC by matching the source and target distributions underlying multiple task-specific layers of the deep network using optimal multi-kernel two-sample matching method [38], [13]. With DAN-ent features in Figure 3(d), the target categories are aligned much better with the source categories, which makes the source classifier better generalizable to the target domain. (4) Finally, we can see that even with DAN-ent features, the target points are still not sufficiently separable, i.e. the decision boundary has to pass through the high-density region to make predictions, which violates the low-density separation criterion [17]. As shown in Figure 3(e), DAN learns the most transferable and discriminative features by addressing these above problems.

5.4.2 Case Study

To enable a more concrete understanding about which kind of errors the algorithms are making, we provide a case study by showing sample images where our method VDAN (ME) succeeds while the strongest competitor RevGrad fails and vice versa. The sample images along with their ground truth labels are illustrated in Figure 4, where the images classified simultaneously right or wrong by both methods are omitted. It is interesting that the number of classes that RevGrad fails but VDAN (ME) succeeds is 16, while this number for vice versa is only 11. This testifies that VDAN (ME) is more robust than RevGrad to diverse variations in image classes. We have to note that most failure predictions are made for images of complex backgrounds or of similar appearances across different classes (e.g. "stapler" and "tape dispenser").

5.4.3 Proxy-A-Distance

The theory of domain adaptation [12], [2] suggests the $\mathcal{A}$-distance as a measure of cross-domain discrepancy, which, together with the source risk, will bound the target risk. As computing the exact $\mathcal{A}$-distance is intractable, the proxy $\mathcal{A}$-distance (PAD) 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 of distinguishing the input samples between the source and target domains. Figure 5(a) shows the PADs on task $\mathbf{A} \rightarrow \mathbf{W}$ (10 classes) with features of CNN, DDC and (V)DAN based on AlexNet (left bars) and GoogLeNet (right bars), respectively. Since deep features can be discriminative both for classifying different categories and different domains [8], they may deteriorate domain adaptation by enlarging the cross-domain discrepancy [12]. It is very desirable that the PADs on DDC and (V)DAN features are much smaller than the PADs on CNN features, which guarantees more transferable features. Another interesting result is that the PADs of GoogLeNet-based methods (22 layers) are even slightly larger than the PADs of AlexNet-based methods (8 layers). This confirms our main motivation that the features of higher task-specific layers in very deep networks are also not safely transferable.



(a) Proxy-$\mathcal{A}$-Distance (b) MMD Penalty λ (c) Entropy Penalty γ (d) Algorithm Scalability

Fig. 5. Empirical analysis: (a) Proxy-$\mathcal{A}$-Distance of different features; (b)–(c) Sensitivity of λ and γ (dashed lines show best baselines); (d) Scalability.

5.4.4 Parameter Sensitivity

The DAN and VDAN models have two hyper-parameters: the MMD penalty λ and entropy penalty γ. We conduct parameter sensitivity analysis on transfer tasks $\mathbf{A} \rightarrow \mathbf{W}$ (31 classes) and $\mathbf{C} \rightarrow \mathbf{W}$ (10 classes). When testing a specific parameter, we fix the other parameters and vary the parameter of interest in $\{0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$. The results are demonstrated in Figures 5(b) and 5(c), with the best results of the baseline methods shown as dashed lines. We observe that the accuracy of DAN and VDAN first increases and then decreases as λ, γ vary and shows stable accuracy curves, which can consistently outperform the best baseline.

5.4.5 Algorithm Scalability

Finally, we empirically testify that the proposed models can scale linearly to large sample size and achieve comparable computational cost with the standard deep networks such as AlexNet and GoogLeNet. Table 5(d) shows that the running times of DAN and VDAN are roughly $1.2\times$ that of AlexNet and GoogLeNet, respectively. This highlights the prospect of the proposed models to big domain adaptation applications.

6 Conclusion

This paper presented a novel learning framework for deep adaptation networks to enhance feature transferability from all task-specific layers of deep convolutional networks. We confirm that while general features can generalize well to a novel target task, specific features tailored to an original task cannot bridge the domain discrepancy effectively. We show that the feature transferability can be enhanced substantially by matching kernel embeddings of multi-layer representations across domains in reproducing kernel Hilbert spaces. The very deep architectures and the low-density separation of target-unlabeled data substantially promote the feature transferability, while an optimal multi-kernel learning strategy and new discrepancy with distinguishable test locations improve the effectiveness of distribution matching. The unbiased estimate of the kernel embedding leads to an efficient linear-time algorithm. Comprehensive empirical evaluation on standard domain adaptation datasets verifies the efficacy of very deep adaptation networks against the state of the art.

Acknowledgments

The authors would like to thank Yue Cao, Baochen Sun and Yuchen Zhang for helpful discussions and suggestions. This work is supported by National Natural Science Foundation of China (61502265, 61325008), National Key R&D Program of China (2016YFB1000701, 2015BAF32B01), and Key Project of Tsinghua TNList Laboratory and NEL-BDSS Laboratory.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng. (TKDE)*, vol. 22, no. 10, pp. 1345–1359, 2010.
[2] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Conf. on Learn. Theory (COLT)*, 2009.
[3] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw. Learn. Sys. (TNNLS)*, vol. 22, no. 2, pp. 199–210, 2011.
[4] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Int. Conf. Comput. Vis. (ICCV)*, 2013.
[5] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Int. Conf. Mach. Learn. (ICML)*, 2013.
[6] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Int. Conf. Mach. Learn. (ICML)*, 2013.
[7] X. Wang and J. Schneider, "Flexible transfer learning under support and model shift," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014.
[8] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Int. Conf. Mach. Learn. (ICML)*, 2011.
[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Int. Conf. Mach. Learn. (ICML)*, 2014.
[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014.
[11] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2007.
[12] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn. (MLJ)*, vol. 79, no. 1-2, pp. 151–175, 2010.
[13] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu, "Optimal kernel choice for large-scale two-sample tests," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2012.
[14] K. P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton, "Fast two-sample testing with analytic representations of probability measures," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1981–1989.
[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, June 2015.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016.
[17] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 529–536.
[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," arXiv:1409.0575, Tech. Rep., 2014.
[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *IEEE Conf. on Comput. Vis. and Pattern Recognition (CVPR)*, June 2013.

[20] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, "LSDA: Large scale detection through adaptation," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014.
[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. Comput. Vis. and Pattern Recognition (CVPR)*, 2014.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2012.
[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia*, 2014.
[24] M. Long, J. Wang, Y. Cao, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Int. Conf. Mach. Learn. (ICML)*, 2015.
[25] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 36, no. 6, pp. 1134–1148, 2014.
[26] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 34, no. 3, pp. 465–479, 2012.
[27] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. Comput. Vis. (ECCV)*, 2010.
[28] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE Int. Conf. on Comput. Vis. (ICCV)*. IEEE, 2011.
[29] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conf. Comput. Vis. and Pattern Recognition (CVPR)*, 2012.
[30] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Simultaneous deep transfer across domains and tasks," in *Int. Conf. Comput. Vis. (ICCV)*, 2015.
[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res. (JMLR)*, vol. 12, pp. 2493–2537, 2011.
[32] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI Conf. Art. Intell. (AAAI)*, 2016.
[33] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," in *NIPS 2014 Workshop on Transfer and Multi-task learning: Theory Meets Practice*, 2014.
[34] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.
[35] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Int. Conf. Mach. Learn. (ICML)*, 2011.
[36] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," arXiv:1412.3474, Tech. Rep., 2014.
[37] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Mach. Learn. (ICML)*, 2015.
[38] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res. (JMLR)*, vol. 13, pp. 723–773, Mar. 2012.
[39] W. Jitkrittum, Z. Szabó, K. P. Chwialkowski, and A. Gretton, "Interpretable distribution features with maximum testing power," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 181–189.
[40] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, G. Lanckriet, and B. Schölkopf, "Kernel choice and classifiability for rkhs embeddings of probability distributions." in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2009.
[41] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, "Equivalence of distance-based and rkhs-based statistics in hypothesis testing," *The Annals of Statistics*, vol. 41, no. 5, pp. 2263–2291, 2013.
[42] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. MIT press Cambridge, 2006.
[43] M. Long, J. Wang, J. Sun, and P. S. Yu, "Domain invariant transfer kernel learning," *IEEE Trans. Knowl. Data Eng. (TKDE)*, vol. 27, no. 6, pp. 1519–1532, 2015.
[44] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep., 2007.
[45] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *IEEE Conf. Comput. Vis. and Pattern Recognition (CVPR)*, 2011.
[46] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE Int. Conf. on Comput. Vis. (ICCV)*, 2013, pp. 2960–2967.
[47] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren, "Cross validation framework to choose amongst models and datasets for transfer learning," in *ECML/PKDD*. Springer, 2010, pp. 547–562.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Representations (ICLR), 2015 (arXiv:1409.1556v6)*, 2015.

Mingsheng Long received the B.E. degree in Electrical Engineering and the Ph.D. degree in Computer Science in 2008 and 2014 respectively, both from Tsinghua University. He is an assistant professor in the School of Software, Tsinghua University. He was a visiting researcher in the AMPLab, UC Berkeley from 2014 to 2015. His research interests include machine learning, computer vision, and big data analytics.

Zhangjie Cao is pursuing the B.E. degree in Computer Software from Tsinghua University, China. His research interests include machine learning and computer vision.

Jianmin Wang graduated from Peking University, China in 1990, and received the M.E. and Ph.D. degrees in Computer Software from Tsinghua University, China in 1992 and 1995, respectively. He is a full professor in the School of Software, Tsinghua University. His research interests include big data management systems, workflow and BPM technology, and large-scale data analytics. He led to develop a product data & lifecycle management system, which has been deployed in hundreds of enterprises in China. He is leading to develop a big data management system named LaUDMS.

Han Zhu received the B.E. degree in Computer Software from Tsinghua University, China in 2014. He is pursuing the M.E. degree in Computer Software at Tsinghua University. His research interests include machine learning and computer vision.

Michael I. Jordan is the Pehong Chen Distinguished Professor in the Department of Electrical Engineering and Computer Science and the Department of Statistics at the University of California, Berkeley. His research interests bridge the computational, statistical, cognitive and biological sciences, and have focused in recent years on Bayesian nonparametric analysis, probabilistic graphical models, spectral methods, kernel machines and applications to problems in distributed computing systems, natural language processing, signal processing and statistical genetics. Prof. Jordan is a member of the National Academy of Sciences, a member of the National Academy of Engineering and a member of the American Academy of Arts and Sciences. He received the David E. Rumelhart Prize in 2015 and the ACM/AAAI Allen Newell Award in 2009. He is a Fellow of the AAAI, ACM, ASA, CSS, IEEE, IMS, ISBA and SIAM.