Deep Learning of Transferable Representation for Scalable Domain Adaptation

Mingsheng Long, Jianmin Wang*, Yue Cao, Jiaguang Sun, and Philip S. Yu, *Fellow, IEEE*

Abstract—Domain adaptation generalizes a learning model across source domain and target domain that are sampled from different distributions. It is widely applied to cross-domain data mining for reusing labeled information and mitigating labeling consumption. Recent studies reveal that deep neural networks can learn abstract feature representation, which can reduce, but not remove, the cross-domain discrepancy. To enhance the invariance of deep representation and make it more transferable across domains, we propose a unified deep adaptation framework for jointly learning transferable representation and classifier to enable scalable domain adaptation, by taking the advantages of both deep learning and optimal two-sample matching. The framework constitutes two inter-dependent paradigms, unsupervised pre-training for effective training of deep models using deep denoising autoencoders, and supervised fine-tuning for effective exploitation of discriminative information using deep neural networks, both learned by embedding the deep representations to reproducing kernel Hilbert spaces (RKHSs) and optimally matching different domain distributions. To enable scalable learning, we develop a linear-time algorithm using unbiased estimate that scales linearly to large samples. Extensive empirical results show that the proposed framework significantly outperforms state of the art methods on diverse adaptation tasks: sentiment polarity prediction, email spam filtering, newsgroup content categorization, and visual object recognition.

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

1 Introduction

The generalization performance of supervised learning with insufficient training data will be unsatisfactory for practical applications, while manual labeling of sufficient training data for all application domains may be prohibitive. Domain adaptation has been established to be effective in reducing human labeling efforts by leveraging rich labeled data from other relevant domains [1]. The challenge is that different domains usually follow substantially different distributions, which has posed a major bottleneck for adapting predictive models across domains. For example, a sentiment classifier [2] built for one product reviews may not predict well the polarity of reviews to another product, since different words can be used to express sentiment in different domains, e.g. words like "blur", "fast", "sharp" are used to comment the *electronics* products, while they do not carry sensible opinion in the *books* products [3]. For another 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. In such cases, effective domain adaptation methods are highly desirable to reduce domain discrepancy and labeling consumption.

The domain adaptation problems involve two types of datasets, one from a source domain and the other from a target domain. The *source domain* contains sufficient amount of labeled data such that a classifier can be reliably built. The *target domain* contains large amount of unlabeled data that follows a substantially different but related distribution. The goal is to correct the distribution mismatch such that a supervised classifier can be effectively transferred across different domains. This poses two key challenges to enable domain adaptation: (1) how to reduce the domain discrepancy, and (2) how to learn transferrable feature representation.

A fruitful stream of previous works have been devoted to address the first challenge, typically by minimizing the Maximum Mean Discrepancy (MMD) [4], a nonparametric statistic that measures the distribution discrepancy in terms of the distance between the kernel mean embeddings of the source and target data. The main objective is to identify a feature representation or instance weighting through which the distribution discrepancy (MMD) is formally reduced [5], [6], [7], [8], [9], [10]. However, MMD is a kernel method that may be restricted by several disadvantages. (1) The kernel functions only characterize local generalization, which is ineffective for capturing global nonlinearity of data to embody the distribution discrepancy [11]. (2) A predefined kernel is not optimal for maximizing the two-sample matching power of MMD [12], while how to learn the optimal kernel is nontrivial. (3) Kernel methods often scale quadratically to the number of samples, which prohibits their applications to big data problems. These open issues are jointly important for robust domain adaptation from large-scale dataset, while they have not been well addressed in previous works.

With the recent revolution of deep learning [11], it has been successfully applied to extract abstract representation for domain adaptation [13], [14], [15], [16]. Deep learning is able to disentangle hidden factors that explain variations underlying the data samples, and hierarchically group features in accordance with their relatedness to the invariant factors [13]. This establishes transfer across domains as the

- *M. Long, J. Wang, Y. Cao, and J. Sun are with the School of Software, Tsinghua TNList Lab for Info. Sci. and Tech., Tsinghua University, Beijing, China. E-mail: mingsheng@tsinghua.edu.cn, jimwang@tsinghua.edu.cn, yue-cao14@mails.tsinghua.edu.cn, sunjg@tsinghua.edu.cn. Corresponding author: J. Wang.*
- *P. S. Yu is with the Institute for Data Science, Tsinghua University, and with the University of Illinois at Chicago, IL, USA. E-mail: psyu@uic.edu.*

deep features contain abstract concepts that are invariant to domain-specific distributions. For example, in the *electronics* domain, domain-specific words like "blur", "fast", "sharp" should reconstruct, and be reconstructed by, co-occurring domain-shared features, typically of similar sentiment (e.g. "good" or "love"). Hence, the source-trained classifier can assign weights to features that even never occur under the original feature representation [15]. However, disentangling the hidden factors of variations may unexpectedly enlarge the cross-domain distribution discrepancy, as the domains represented by new deep features will become more "compact" and more mutually distinguishable. While the invariant latent factors exploited by deep learning can suppress domain-specific variations and improve generalization, the increased cross-domain discrepancy may reversely deteriorate domain adaptation performance, leading to statistically unbounded target error [17], [18], [19]. This problem has not been addressed for general-purpose deep learning methods.

Inspired by the literature's latest understanding on deep neural networks for learning compact and invariant feature representations, in this paper, we propose a unified deep adaptation framework to jointly learn transferable representation and classifier to enable scalable domain adaptation, by taking the advantages of both deep learning and optimal two-sample matching. The framework constitutes two interdependent paradigms, *unsupervised pre-training* for effective training of deep models using deep denoising autoencoders [20], and *supervised fine-tuning* for effective exploitation of discriminative information using deep neural networks [21]. For the unsupervised pre-training paradigm, we propose a *Transfer Denoising Autoencoder* (TDA) model, where the learned representations of multiple autoencoders are embedded to reproducing kernel Hilbert spaces (RKHSs) and the mean embeddings of different domain distributions are formally matched. For the supervised fine-tuning paradigm, we propose a *Transfer Deep Network* (TDN) model, which is constructed by stacking the encoders parts unfolded from the pre-trained TDAs, and is fine-tuned using a supervised classifier at the output layer. As the effectiveness of two-sample matching based on MMD [4] is restricted by the local generalization issue [11] and suboptimal kernel issue [12], we propose a multiple kernel method that learns optimal kernel for two-sample matching. To enable scalable adaptation to large-scale applications, we develop a linear-time algorithm using B-test, an unbiased estimate of MMD [22] that scales linearly to large-scale samples. Extensive empirical evidence shows that the proposed models significantly outperform state of art methods on diverse adaptation tasks: sentiment polarity prediction, email spam filtering, newsgroup content categorization, and visual object recognition. The contributions of this paper are summarized as follows.

- A deep adaptation framework is proposed for robust domain adaptation, taking advantages from both unsupervised pre-training and supervised fine-tuning.
- A multi-kernel learning method is devised for two-sample matching of different domain distributions.
- A linear-time learning algorithm is devised to enable scalable deep domain adaptation based on MMD.
- A rigorous theoretical analysis of generalization error bound is provided to establish statistical guarantees.

The remainder of this paper is organized as follows. We begin by reviewing the related works in Section 2. We present the proposed deep adaptation models in Section 3, and derive the learning algorithms and theoretical analysis in Section 4. Empirical evaluations are reported in Section 5, while conclusion and future work are enclosed in Section 6.

2 Related Work

The performance of supervised learning machines depend on training data, while it is usually time-consuming to collect sufficient training data. Domain adaptation is a general learning paradigm that allows classification algorithms to leverage rich labeled data from relevant domains. It has been widely deployed to save the manual-labeling efforts in many areas, e.g. machine learning [5], [6], [8], [23], [10], [24], data mining [25], [26], natural language processing [2], [3], [27], and computer vision [28], [29], [9], etc. Prior domain adaptation methods can be generally put into two categories [1]: (1) instance weighting [5], [27], which selects the source instances that are the most relevant to the target domain; (2) feature extraction [2], [3], [6], [29], [8], [23], which learns a shallow feature representation that remains invariant across domains. However, without learning deep features that can suppress domain-specific exploratory factors, such feature invariance may be limited by domain-specific structures.

Deep learning extracts representation that disentangles and hides more or less the explanatory factors of variation underlying data samples [11]. Such representation manifests invariant factors underlying different populations and can successfully establish domain adaptation [13], [15], [16], [30]. Deep learning has also been extended to multimodal and multi-source problems [14]. However, these methods rely on the assumption that deep learning can successfully learn the desired transferable representations for domain adaptation. As we will clarify by both theoretical and empirical results, the domain discrepancy poses a general bottleneck to the generalization performance of machine learning algorithms that cannot be tackled solely by deep learning methods.

The domain discrepancy should be reduced to achieve lower transfer errors [17], [18], [19]. Several parallel works [31], [32] add an adaptation layer to the deep convolutional neural network (CNN), but may be restricted by two defects: (1) they only adapt a single layer of the network, which may be ineffective since there are multiple layers where hidden features are not directly transferable [16]; (2) they either use suboptimal kernel for MMD-based two-sample matching, or inefficient adversarial training for the source-target discrimination, which further degrades the adaptation effectiveness [12], [33]. While the deep adaptation network (DAN) model proposed by Long et al. [33] gives state of the art results, it is restricted to visual domains and is inapplicable to textual domains. Finally, the proposed framework contrasts clearly from a concurrent work on supervised representation learning with deep autoencoders (TLDA) [34]: (1) TDN is deep neural network while TLDA is limited to only one hidden layer and one classifier layer; (2) TDA and TDN learn optimal kernel to maximize two-sample matching power while TLDA is limited to mean matching with KL-divergence; (3) TDA and TDN have rigorous generalization error bounds while it is still unclear whether TLDA has such guarantees.

TABLE 1
Notations and Their Descriptions Frequently Used in This Paper

Notation	Description	Notation	Description
$\mathcal{D}_s, \mathcal{D}_t$	source, target	$\mathbf{x}, \tilde{\mathbf{x}}$	input: original, corrupted
d, n	#features, #samples	$\mathbf{z}, \tilde{\mathbf{z}}$	activation: original, corrupted
c	corruption probability	$\mathbf{W}, \mathbf{b}$	deep network parameters
λ	adaptation penalty	$f_\theta, g_{\theta'}$	encoder, decoder
$k, \boldsymbol{\beta}$	multi-kernel function	f_θ^{sup}	supervised classifier: softmax

3 Scalable Domain Adaptation Networks

This section presents two novel deep learning architectures for scalable domain adaptation. We begin by introducing the multi-kernel maximum mean discrepancy (MK-MMD), a nonparametric test statistic for optimal two-sample distribution comparison. Then we propose two interplay architectures: the transfer denoising autoencoder (TDA) model that learns transferable representation through pre-training on unlabeled data, and the transfer deep network (TDN) model that learns both transferable representation and classifier through fine-tuning on labeled and unlabeled data. Both pre-training and fine-tuning are vital for effective learning of deep models: pre-training mitigates the trap of local minima and fine-tuning exploits the supervised information.

In unsupervised domain adaptation, we have a *source* domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled examples, and a *target* domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ with n_t unlabeled examples. The source domain and target domain follow different probability distributions p and q, respectively. We do not assume that all domains should share identical features and we will pad all input vectors with zeros to make both domains be of equal dimensionality d. We are targeting a deep architecture which is able to learn *transferable* representation to bridge the domain discrepancy, and construct a classifier $y = h(\mathbf{x})$ which minimizes target risk $R_{\mathcal{D}_t}(h) = \Pr_{(\mathbf{x},y)\sim q}[h(\mathbf{x}) \neq y]$ using source supervision. When multiple source domains $\{\mathcal{D}_s\}_{s=1}^S$ are available, we can extend the deep architectures for multiple-source domain adaptation. Furthermore, when there is a small amount of labeled examples available in the target domain, the transferable representation learned by our method can be fed to existing semi-supervised methods to enable semi-supervised domain adaptation. The frequent notations and their descriptions are summarized in Table 1.

3.1 Multi-Kernel Maximum Mean Discrepancy

The challenge of domain adaptation is that target domain has no labeled information. To approach this problem, many existing methods bound the target error by the source error plus a discrepancy metric between the source and the target [18]. Two classes of statistics have been explored for the *two-sample* testing, which makes acceptance or rejection decision to the null hypothesis $p = q$, given two samples generated respectively from p and q: Energy Distance (ED) and *Maximum Mean Discrepancy* (MMD) [35]. In this paper, we will focus on the *multiple-kernel* variant of MMD (MK-MMD) proposed by [12], which is formalized to jointly maximize the two-sample test power and minimize the Type II error, i.e. failure of rejecting the false null hypothesis $p = q$.

Let $\mathcal{H}_k$ be the reproducing kernel Hilbert space (RKHS) induced by a characteristic kernel k. The *kernel mean embedding* of distribution p in $\mathcal{H}_k$ is a unique function $\mu_k(p)$ which satisfies $\mathbb{E}_{\mathbf{x}\sim p} f(\mathbf{x}) = \langle f(\mathbf{x}), \mu_k(p)\rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$. The kernel k is *characteristic* if the kernel mean embedding $\mu_k(p)$ is injective, and thus each distribution can be uniquely represented in the RKHS and all statistical features of distributions are preserved by the kernel 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. This is advantageous as the kernel mean embedding has dimension-independent rates of convergence, while rates of convergence for many density estimation procedures are dependent on the input dimension [4]. The multi-kernel variant of maximum mean discrepancy (MK-MMD) [12] between distributions p and q is defined as the RKHS-distance between $\mu_k(p)$ and $\mu_k(q)$,

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

where $\phi(\cdot)$ is the nonlinear feature mapping that induces $\mathcal{H}_k$. The most important property is that $p = q$ iff $d_k^2(p,q) = 0$ [4]. The multi-kernel $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ is defined as the convex combination of m characteristic kernels $\{k_u\}$,

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

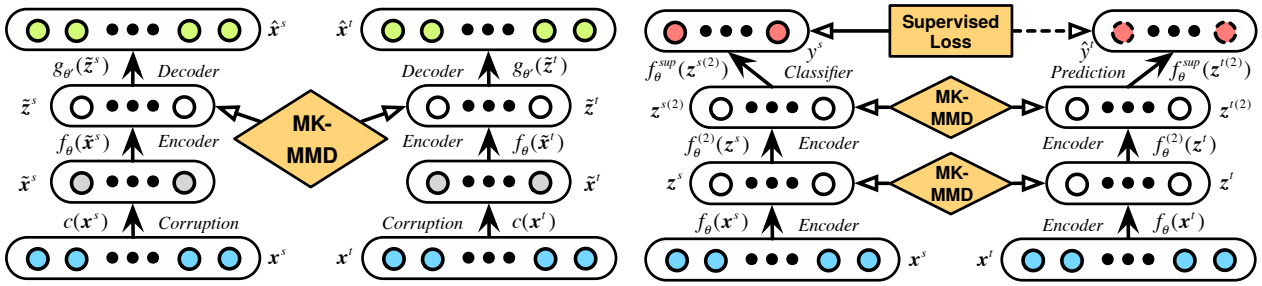
where the constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that the composed multi-kernel k is characteristic. As theoretically studied in [12], the kernel adopted for mean embeddings of p and q is critical in ensuring high test power and low test error, i.e. one should minimize the chance of degenerated two-sample test cases $d_k^2(p,q) \to 0$ when $p \neq q$. The multi-kernel k can leverage multiple kernels to match the moments statistics of distributions at different scales. By minimizing the test error, we can establish an optimal kernel selection method for powerful two-sample matching.

A successful strategy to control the domain discrepancy is to find an invariant feature representation through which the source domain and target domain become similar [17], [18], [19]. Note that MMD has been extensively explored in this line of works [5], [6], [7], [8], [9], [10], [36]. However, to date there has been no attempt learning both transferable representation and classification model through MK-MMD in deep networks. Hence, prior shallow learning methods may be restricted by representation weakness as they cannot disentangle the exploratory factors of variation underlying data samples [11], while prior deep learning methods may be restricted by adaptation weakness as they cannot correct the distribution mismatch using two-sample matching [12]. These problems motivate powerful deep architectures that seamlessly integrate optimal two-sample matching module.

3.2 Transfer Denoising Autoencoder

In this subsection, we present *Transfer Denoising Autoencoder* (TDA), an unsupervised model for learning from large-scale unlabeled data based on denoising autoencoders [20], which explores the idea of optimal two-sample matching to learn transferable representation for scalable domain adaptation. TDA enables domain adaptation by unsupervised layer-wise pre-training, which serves both as an effective initialization and adaptive regularization for training deep neural networks [37], [38]. Unsupervised pre-training is important for domain adaptation since labeled data in this scenario is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2554549, IEEE Transactions on Knowledge and Data Engineering



(a) Transfer Denoising Autoencoder (TDA) (b) Transfer Deep Network (TDN)

Fig. 1. The proposed deep architectures for scalable domain adaptation: (a) transfer denoising autoencoder (TDA) that learns transferable features via unsupervised pre-training; and (b) transfer deep network (TDN) that learns both transferable features and classifier via supervised fine-tuning.

often insufficient for supervised training of deep networks. Figure 1(a) shows the architecture of the TDA model.

Deep learning [11] is armed with the capability of learning distributed, compositional, and abstract representations for complicated sensory data such as text, image, and video. In this work, we adapt the Denoising Autoencoder (DA) [20] for transfer learning, which has many successful cases especially for text mining and natural language processing. DA is a fully connected neural network comprising one input layer, one output layer, and one or multiple hidden layers, which targets at *denoising* artificially corrupted inputs, i.e. learning to reconstruct clean inputs from *corrupted* versions. It has been shown that training with such a denoising criterion is related to data-dependent adaptive regularization in generalized linear models [39] and marginalized denoising autoencoders [40], which captures the manifold structure of the input distribution to undo the effect of corruption. The reconstruction from DA is a nearby but higher-density point than the corrupted input point, hence DA may learn more robust representation than ordinary autoencoders [20].

Specifically, let $\mathbf{x}_i, \tilde{\mathbf{x}}_i, \hat{\mathbf{x}}_i$ be the original data sample, the corrupted version of $\mathbf{x}_i$ generated by pre-defined corruption probability $c(\tilde{\mathbf{x}}|\mathbf{x})$, and the reconstructed version of $\mathbf{x}_i$ that is decoded from DA, respectively. Then DA denoises input corruptions by minimizing the reconstruction error of data:

$$\begin{aligned} &\min_{\theta,\theta'} \sum_{i=1}^{n} \mathbb{E}_{c(\tilde{\mathbf{x}}|\mathbf{x})} J\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right) \\ &\tilde{\mathbf{z}}_i = f_\theta\left(\tilde{\mathbf{x}}_i\right) \text{ and } \hat{\mathbf{x}}_i = g_{\theta'}\left(\tilde{\mathbf{z}}_i\right), \end{aligned} \tag{3}$$

where $f_\theta(\cdot)$ and $g_{\theta'}(\cdot)$ are the encoder and decoder respectively, with θ and θ' being the autoencoder parameters, $\tilde{\mathbf{z}}_i$ is the hidden representation of the corrupted input sample $\tilde{\mathbf{x}}_i$, and $J(\cdot,\cdot)$ is the loss function, which can be taken as squared loss $J\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right) \triangleq \left\|\mathbf{x}_i - \hat{\mathbf{x}}_i\right\|^2$ or cross-entropy loss $J\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right) \triangleq -\sum_j \left[x_{ij} \log \hat{x}_{ij} + (1 - x_{ij}) \log\left(1 - \hat{x}_{ij}\right)\right]$. The choices of loss functions will depend on the types of encoder and decoder, which are parametrized respectively as follows

$$\begin{aligned} f_\theta\left(\tilde{\mathbf{x}}_i\right) &= a\left(\mathbf{W}\tilde{\mathbf{x}}_i + \mathbf{b}\right) \\ g_{\theta'}\left(\tilde{\mathbf{z}}_i\right) &= a'\left(\mathbf{W}'\tilde{\mathbf{z}}_i + \mathbf{b}'\right), \end{aligned} \tag{4}$$

where $\theta \triangleq \{\mathbf{W}, \mathbf{b}\}$ and $\theta' \triangleq \{\mathbf{W}', \mathbf{b}'\}$ are the weight and bias terms of the encoder and decoder respectively, a and a' are activation functions, which can be set as linear function $a(x) = x$, rectifier linear unit (ReLU) $a(x) = \max(0, x)$, sigmoid function $a\left(x\right) = \frac{1}{1+e^{-x}}$, or hyperbolic tangent (tanh) function $a\left(x\right) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. When the decoder activation a' is sigmoid and the input $\mathbf{x}$ is binary, the cross-entropy loss is preferred; otherwise the squared loss should be used [20]. We adopt sigmoid activation for the encoder and decoder, and cross-entropy loss for the objective, except for the first layer where we adopt linear activation for the decoder, and squared loss for the objective since our data is real-valued. A featured ingredient of DA is the expectation $\mathbb{E}_{c(\tilde{\mathbf{x}}|\mathbf{x})}(\cdot)$ in Equation (3) that averages over corrupted samples $\tilde{\mathbf{x}}_i$ drawn from corruption process $c\left(\tilde{\mathbf{x}}_i|\mathbf{x}_i\right)$. Computing this expectation exactly is intractable due to the nonlinear encoding and decoding functions f_θ and $g_{\theta'}$. In practice, Equation(3) is optimized by mini-batch stochastic gradient descent (SGD), where the gradient is estimated by drawing a few corrupted versions of $\mathbf{x}_i$ at each iteration. In this paper, we focus on the *masking* corruption $c\left(\tilde{\mathbf{x}}_i|\mathbf{x}_i\right)$ which corrupts each $\mathbf{x}_i$ by random feature removal, i.e. each feature is independently set to 0 with probability $c \in [0, 1]$. This masking corruption proves effective for general-purpose problems [20], [13].

It has been shown that the hidden representation $\tilde{\mathbf{z}}_i = f_\theta\left(\tilde{\mathbf{x}}_i\right)$ learned by DA can disentangle the exploratory factors of variations underlying the sample distributions [13], [15]. However, the latest literature findings also reveal that the deep representation can reduce, but not remove, the distribution discrepancy across the source and target domains [16], [33]. In this paper, we pre-train DA by requiring the distributions of the source and target become similar under the hidden representation. This is realized by substituting the hidden representation to the MK-MMD formulation (1)

$$\min_{\theta,\theta'} \max_{k \in \mathcal{K}} d_k^2\left(\mathcal{D}_s^z, \mathcal{D}_t^z\right) = \left\|\mathbb{E}_p \phi\left(\tilde{\mathbf{z}}^s\right) - \mathbb{E}_q \phi\left(\tilde{\mathbf{z}}^t\right)\right\|_{\mathcal{H}_k}^2, \tag{5}$$

where $\mathbf{x}^s, \tilde{\mathbf{z}}^s \sim p$ and $\mathbf{x}^t, \tilde{\mathbf{z}}^t \sim q$ are the source sample and target sample generated from distributions p and q, respectively, while $\mathcal{D}_s^z \triangleq \{\tilde{\mathbf{z}}_i^s\}_{i=1}^{n_s}$ is the hidden representation of source data, and $\mathcal{D}_t^z \triangleq \{\tilde{\mathbf{z}}_i^t\}_{i=1}^{n_t}$ is the hidden representation of target data. To enable generalization of DA across source and target $\mathcal{D}_s$ and $\mathcal{D}_t$, we integrate MK-MMD (5) as an adaptation regularizer to the DA empirical risk (3), leading to the proposed *Transfer Denoising Autoencoder* (TDA) model:

$$\min_{\theta,\theta'} \max_{k \in \mathcal{K}} \sum_{i=1}^{n} \mathbb{E}_{c(\tilde{\mathbf{x}}|\mathbf{x})} J\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right) + \lambda \left\|\mathbb{E}_p \phi\left(\tilde{\mathbf{z}}^s\right) - \mathbb{E}_q \phi\left(\tilde{\mathbf{z}}^t\right)\right\|_{\mathcal{H}_k}^2, \tag{6}$$

where $\lambda > 0$ is the penalty parameter of adaptation regularization. By minimizing MK-MMD for learning the nonlinear

feature representation, TDA is capable to learn distributed, compositional, and abstract representation which manifests invariant structures across domains. An illustration of TDA is shown in Figure 1(a). After the first layer TDA is trained, we can construct a *deep* architecture by stacking multiple TDAs on its top in a layerwise way [20], where the hidden representation of the lower-layer TDA, i.e. $f_\theta(\mathbf{x}_i)$, is used as the input of the upper-layer TDA. It is noteworthy that input corruption is only used for denoising-training of DA so that it can learn useful features. Once the encoder f_θ has been learned, it will be applied on uncorrupted inputs to produce hidden representation $f_\theta(\mathbf{x}_i)$. We train the Stacked TDA in a greedy layerwise manner using mini-batch stochastic gradient descent (SGD) [20]. Through stacking, the hidden representations are made more abstract to disentangle the exploratory factors of variations underlying data samples, and more invariant to reduce the distribution discrepancy across domains to enable effective domain adaptation.

Marginalized TDA It has been shown that DA [20] has high computational cost, because the nonlinear activation functions $a(\cdot)$ and $a'(\cdot)$ make its optimization problem non-convex while its random feature corruption is made on each small-batch of input samples. Inspired by the neat idea of marginalizing linear denoising autoencoders (mSDA) [15], where the random feature corruption is marginalized out by taking the expectation of random corruptions, we speed up TDA by applying the same *marginalization* manipulation. Conceptually, the marginalization is equivalent to training the autoencoders with an infinite number of the corrupted input samples. The marginalization is applicable only after simplifying the TDA model (6) using the *linear* activation function $a(x) = x$ and $a'(x) = x$. This marginalized version of TDA is named as mTDA, whose attractive advantage is the capability of learning from infinite corruptions and performing much faster than TDA. As the nonlinearity and the deep architecture are arguably the two key contributors to the breakthrough of deep learning [11], to still benefit from nonlinearity, we inject the nonlinearity using $a(\cdot) = \tanh(\cdot)$ after the linear hidden representation $f_\theta(\mathbf{x}_i)$ is computed. To perform the layer-wise training, several mTDA layers are stacked by feeding the output of the $(\ell-1)^{\text{th}}$ mTDA (after the activation function) as the input to the ℓ^{th} layer mTDA. Empirical evidence shows that marginalized autoencoders are mainly effective for text mining applications [15], [41]. A distinction is that mSDA only uses decoder $g_{\theta'}$ while mTDA uses both encoder f_θ and decoder $g_{\theta'}$, hence mSDA can only extract representations with the same dimension as input, while mTDA can learn dimension-reduced representations using a bottleneck encoder with fewer network parameters.

3.3 Transfer Deep Network

The composition of multiple levels of nonlinearity in neural networks is key to efficiently model complex relationships across exploratory factors and to achieve better generalization performance on challenging perception tasks [11], [20]. This philosophy has motivated the latest breakthroughs in deep neural networks, such as deep convolutional neural network (CNN) [42] for computer vision and deep recursive neural network (RNN) [43] for natural language processing. Recent process shows that *unsupervised pre-training* as done in TDA, and *supervised fine-tuning* as done in CNN, are both important for the effective learning of deep neural networks, while supervised fine-tuning is very important by adjusting deep networks to better fit the perception task, which serves as the key to recent breakthroughs of both CNN and RNN. Therefore, one major limitation of TDA is that it cannot be effectively adjusted to the supervised learning task due to its unsupervised pre-training paradigm. Another limitation of TDA is that it constructs the deep architecture through "stacking", which is not truly "deep" since its upper-layers cannot influence its lower-layers by back-propagation.

To benefit from the worlds of unsupervised pre-training and supervised fine-tuning, we further propose the *Transfer Deep Network* (TDN) model, a supervised model for learning from both labeled source and unlabeled target data based on multilayer perceptrons [21]. We again explore the idea of optimal two-sample matching to learn *both* transferable representation and classifier for scalable domain adaptation. Supervised fine-tuning is critical for domain adaptation as labeled data can be fully exploited to make the transferable representation more discriminative. TDA serves as an indispensable pre-training phase for TDN, otherwise TDN cannot be trained effectively due to gradient vanishing [37], [38]. Figure 1(b) shows the architecture of the TDN model.

Stacking multiple TDAs to initialize the deep network works in much the same way as stacking multiple denoising autoencoders [20] or ordinary autoencoders [44]. It is noteworthy to specify that input corruption is only used for the denoising-training of each individual TDA so that it may learn useful feature extractors. Once the encoder f_θ has been learned, it will henceforth be applied on uncorrupted inputs to produce the representation that will serve as clean input for training the next TDA. After a stack of TDA encoders has thus been built, a softmax regression classifier can be added on top of the TDA encoders, yielding a deep neural network amenable to supervised learning, which can be fine-tuned to exploit source labeled data and match cross-domain data. We name the resulting architecture as *Transfer Deep Network* (TDN), which is a multilayer perceptron pre-trained by TDA and regularized by the MK-MMDs on all the hidden layers:

$$\min_{\{\theta^\ell\}_1^l} \max_{k\in\mathcal{K}} \sum_{i=1}^{n_s} J\left(h\left(\mathbf{x}_i^s\right), y_i^s\right) + \lambda \sum_{\ell=1}^{l-1} \left\| \mathbb{E}_p \phi(\mathbf{z}^{s,\ell}) - \mathbb{E}_q \phi(\mathbf{z}^{t,\ell}) \right\|_{\mathcal{H}_k}^2, \tag{7}$$

where $\lambda > 0$ is the penalty parameter of adaptation regularization, l is the number of layers, $\mathbf{z}^{s,\ell}$ and $\mathbf{z}^{t,\ell}$ are the ℓ-th layer hidden representation of source sample $\mathbf{x}^s$ and target sample $\mathbf{x}^t$ respectively, $h(\mathbf{x}_i^s)$ is the classifier produced by the deep neural network, and $J(\cdot,\cdot)$ is the cross-entropy loss function for the softmax regression that is defined as follows

$$J\left(h\left(\mathbf{x}_i^s\right), y_i^s\right) = -\sum_{j=1}^{c} 1\left\{y_i^s = j\right\} \log h_j\left(\mathbf{x}_i^s\right), \tag{8}$$

where c is the number of categories in source domain, and $h_j\left(\mathbf{x}_i^s\right) = e^{z_{ij}^{s,l}} / \sum_{j'} e^{z_{ij'}^{s,l}}$ is the softmax function computing the probability of predicting sample $\mathbf{x}_i^s$ to the j-th category. By minimizing MK-MMD in multiple hidden layers $1 \le \ell \le l-1$ during supervised fine-tuning from labeled source data, TDN is capable of learning both transferable representation and classifier to enable effective deep domain adaptation.

The TDA and TDN models take advantages from *both* unsupervised pre-training for effective training of deep networks, and supervised fine-tuning for exploiting supervised information. The optimal multi-kernel two-sample matching based on the MK-MMD is performed for *both* worlds of methods, leading to an optimal deep learning framework for effective domain adaptation. The formulations of TDA (6) and TDN (7) are *minimax* problems that learn abstract deep representation by minimizing the MK-MMD with respect to network parameters θ, θ', and jointly learn optimal multi-kernel by maximizing the MK-MMD with respect to the kernel parameters $\boldsymbol{\beta}$. We will further show in the theoretical analysis that the TDA and TDN models can achieve a tighter bound for domain adaptation. Another notable thing is that MK-MMD is imposed on *multiple* layers of TDA and TDN, which matches the hierarchical representations at different abstraction levels for a deep consolidation of transferability.

4 Algorithm and Analysis

We present linear-time learning algorithms for the TDA and TDN models, and provide theoretical analysis on the learning bound for scalable domain adaptation. The scalable learning algorithms are based on the low-variance unbiased estimate of MK-MMD [22]. Note that most previous domain adaptation methods requires $O(n^2)$ cost to compute MMD.

4.1 Learning Network Parameters

Using the kernel trick $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, MK-MMD in Equation (1) can be computed as the expectation of kernel functions $d_k^2(p,q) = \mathbb{E}_{\mathbf{x}^s \mathbf{x}'^s} k(\mathbf{x}^s, \mathbf{x}'^s) + \mathbb{E}_{\mathbf{x}^t \mathbf{x}'^t} k(\mathbf{x}^t, \mathbf{x}'^t) - 2\mathbb{E}_{\mathbf{x}^s \mathbf{x}^t} k(\mathbf{x}^s, \mathbf{x}^t)$, 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 mini-batch SGD is important to the effective training of deep neural networks. While prior works based on MMD [6], [31] rarely address this problem, we believe it is more critical for deep learning.

In this paper, we adopt B-test [22], a low-variance and unbiased estimate of MK-MMD, which is defined as follows

$$\begin{aligned} \hat{d}_k^2(\mathcal{D}_s, \mathcal{D}_t) &\triangleq \sum_{b=0}^{n_s/B-1} \sum_{i=bB+1}^{(b+1)B} \sum_{j=bB+1, j\neq i}^{(b+1)B} \frac{\kappa\left(\mathbf{x}_i^s, \mathbf{x}_i^t, \mathbf{x}_j^s, \mathbf{x}_j^t\right)}{n_s(B-1)} \\ &\triangleq \sum_{b=0}^{n_s/B-1} \hat{d}_k^2\left(\mathcal{D}_s^b, \mathcal{D}_t^b\right), \end{aligned} \tag{9}$$

where $\kappa(\mathbf{x}_i^s, \mathbf{x}_i^t, \mathbf{x}_j^s, \mathbf{x}_j^t) \triangleq k(\mathbf{x}_i^s, \mathbf{x}_j^s) + k(\mathbf{x}_i^t, \mathbf{x}_j^t) - k(\mathbf{x}_i^s, \mathbf{x}_j^t) - k(\mathbf{x}_j^s, \mathbf{x}_i^t)$, B is the size of mini-batch used in the mini-batch SGD, and thus n_s/B is the number of mini-batches in each training epoch, and b is the index of mini-batch. B-test can be computed with linear-time cost, i.e. $O(Bn)$, given that in mini-batch SGD the size B is often small, e.g. $B = 100$. Theoretical result [22] shows that B-test in Equation (9) is a low-variance unbiased estimate of MK-MMD in Equation (1): $\hat{d}_k^2(\mathcal{D}_s, \mathcal{D}_t) \xrightarrow{D} \mathcal{N}\left(d_k^2(p,q), \sigma_u^2 n_s^{-1}\right)$, which shows that B-test converges in distribution to a Gaussian with MK-MMD as the mean and $\sigma_u^2 n_s^{-1}$ as the much lower variance. MK-MMD can be formulated as the sum of n_s/B mini-batches, with each mini-batch denoted by $\hat{d}_k^2\left(\mathcal{D}_s^b, \mathcal{D}_t^b\right)$ indexed by b, which is well fitted to the mini-batch SGD algorithm.

When training deep networks including TDA and TDN by mini-batch SGD, we only need to consider the gradient of TDA objective (6) or TDN objective (7) based on each mini-batch $\{\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b}\}$, where $\mathcal{D}_*^{\ell,b} \triangleq \bigcup_{i=bB+1}^{(b+1)B} \mathbf{z}_i^{*,\ell}$ is the mini-batch of the ℓ-th hidden representation. Since B-test takes a summation form that can be readily decoupled into the sum of MK-MMD on each mini-batch, i.e. $\hat{d}_k^2(\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b})$, we only need to compute gradient $\frac{\partial \hat{d}_k^2\left(\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b}\right)}{\partial \theta^\ell}$ of each mini-batch $\{\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b}\}$ with respect to the network parameters $\{\theta^\ell\}_1^l$. Correspondingly, we can compute the gradient of TDA error $\sum_{\mathbf{x}_i^\ell \in \mathcal{D}_s^{\ell,b} \cup \mathcal{D}_t^{\ell,b}} \frac{\partial J\left(\mathbf{x}_i^\ell, \hat{\mathbf{x}}_i^\ell\right)}{\partial \theta^\ell}$ or TDN error $\sum_{\mathbf{x}_i^s \in \mathcal{D}_s^b} \frac{\partial J(h(\mathbf{x}_i^s), y_i^s)}{\partial \theta^\ell}$ for each mini-batch b in the ℓ-th layer of the Stacked TDA or TDN. Hence, to perform a mini-batch update, we compute the gradient of the TDN objective (7) with respect to θ^ℓ as

$$\nabla_{\theta^\ell}^b = \sum_{\mathbf{x}_i^s \in \mathcal{D}_s^b} \frac{\partial J\left(h\left(\mathbf{x}_i^s\right), y_i^s\right)}{\partial \theta^\ell} + \lambda \frac{\partial \hat{d}_k^2\left(\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b}\right)}{\partial \theta^\ell}. \tag{10}$$

We omit the update rule for TDA, which is straightforward. Such a mini-batch SGD can be easily implemented based on the Pylearn2 library [45]. Given kernel k as the linear combination of m Gaussian kernels $\{k_u(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\gamma_u}\}$, the gradient $\frac{\partial \hat{d}_k^2\left(\mathcal{D}_s^{\ell,b}, \mathcal{D}_t^{\ell,b}\right)}{\partial \theta^\ell}$ can be computed for each data pair based on the chain rule of derivatives. For example,

$$\begin{aligned} \frac{\partial k(\mathbf{z}_i^{s,\ell}, \mathbf{z}_j^{t,\ell})}{\partial \mathbf{W}^\ell} = &-\sum_{u=1}^{m} \frac{2\beta_u}{\gamma_u} k_u(\mathbf{z}_i^{s,\ell}, \mathbf{z}_j^{t,\ell}) \times (\mathbf{z}_i^{s,\ell} - \mathbf{z}_j^{t,\ell}) \\ &\times \left(\dot{a}(\mathbf{z}_i^{s,\ell}) \odot \mathbf{z}_i^{s,(\ell-1)} - \dot{a}(\mathbf{z}_j^{t,\ell}) \odot \mathbf{z}_j^{t,(\ell-1)}\right)^\mathsf{T}, \end{aligned} \tag{11}$$

where $\dot{a}$ is the gradient of activation a. In TDA, the ℓ-th layer input is the $(\ell-1)$-th layer hidden output, i.e. $\mathbf{x}_i^\ell = \mathbf{z}_i^{\ell-1}$.

We speed up the marginalized TDA (mTDA) by exploring the interesting marginalization trick used in [15], that is, the expectation over the corruption probability $c(\tilde{\mathbf{x}}|\mathbf{x})$ can be computed analytically. To see this, we formally derive the gradient of mTDA objective, i.e. Equation (6) using linear activation functions with respect to each $\mathbf{x}_i^\ell \in \mathcal{D}_s^{\ell,b} \cup \mathcal{D}_t^{\ell,b}$ as

$$\frac{\partial J\left(\mathbf{x}_i^\ell, \hat{\mathbf{x}}_i^\ell\right)}{\partial \mathbf{W}^\ell} = -2\mathbf{W}'^\mathsf{T}\left(\mathbb{E}\left[\mathbf{x}_i^\ell \tilde{\mathbf{x}}_i^{\ell\mathsf{T}}\right] + \mathbf{W}'\mathbf{W}\mathbb{E}\left[\tilde{\mathbf{x}}_i^\ell \tilde{\mathbf{x}}_i^{\ell\mathsf{T}}\right]\right), \tag{12}$$

where the bias terms are absorbed into the weight terms for notation brevity. We compute the above expectations by a marginalization trick [15]. An off-diagonal entry in $\tilde{\mathbf{x}}_i^\ell \tilde{\mathbf{x}}_i^{\ell\mathsf{T}}$ is uncorrupted if both the two features j and j' "survived" the corruption, which happens with probability $(1-c)^2$; for diagonal entries, this holds with probability $(1-c)$. Denote by $\bar{\mathbf{c}} = [(1-c), \ldots, (1-c), 1]^\mathsf{T} \in \mathbb{R}^{d+1}$ the "survival" vector, where $\bar{\mathbf{c}}_{d+1} = 1$ is the constant feature never corrupted. The expectations in Equation (12) for denoising corruptions are

$$\begin{aligned} &\mathbb{E}_{c(\tilde{\mathbf{x}}|\mathbf{x})}[\mathbf{x}_i^\ell \tilde{\mathbf{x}}_i^{\ell\mathsf{T}}] = \left(\mathbf{x}_i^\ell \mathbf{x}_i^{\ell\mathsf{T}}\right)_{jj'} \bar{\mathbf{c}}_{j'}, \\ &\mathbb{E}_{c(\tilde{\mathbf{x}}|\mathbf{x})}[\tilde{\mathbf{x}}_i^\ell \tilde{\mathbf{x}}_i^{\ell\mathsf{T}}] = \begin{cases} \left(\mathbf{x}_i^\ell \mathbf{x}_i^{\ell\mathsf{T}}\right)_{jj'} \bar{\mathbf{c}}_j \bar{\mathbf{c}}_{j'}, & j \neq j' \\ \left(\mathbf{x}_i^\ell \mathbf{x}_i^{\ell\mathsf{T}}\right)_{jj'} \bar{\mathbf{c}}_j, & j = j' \end{cases} \end{aligned} \tag{13}$$

Algorithm 1: Deep Adaptation Models: TDA and TDN

Input: Data $\mathbf{X}$; corruption level c, penalty λ, #layers l.
Output: Transferable representation $\mathbf{R}$ and classifier h.

/* Unsupervised pre-training with TDA */
Initialize $\mathbf{X}^1 \leftarrow \mathbf{X}$ and parameters $\{\theta, \theta'\}$ randomly.
for layer $\ell = 1$ **to** $l-1$ **do**
  for epoch $t = 1$ **to** T **do**
    /* Feed-forward pass omitted */
    for mini-batch $b = 1$ **to** n_s/B **do**
      Update $\theta^\ell, \theta'^\ell$ of ℓ^{th} TDA by SGD (10) (12).
    Update $\boldsymbol{\beta}^\ell$ of the ℓ^{th} TDA by QP (15).
  Compute hidden representation $\mathbf{Z}^\ell = f_{\theta^\ell}(\mathbf{X}^\ell)$.
  Set the input of the $(\ell+1)^{\text{th}}$ TDA as $\mathbf{X}^{\ell+1} \leftarrow \mathbf{Z}^\ell$.
/* Supervised fine-tuning with TDN */
Initialize TDN parameters $\{\theta^\ell\}_1^l$ by TDA weights.
for epoch $t = 1$ **to** T **do**
  /* Feed-forward pass omitted */
  for layer $\ell = l$ **to** 1 **do**
    for mini-batch $b = 1$ **to** n_s/B **do**
      Update θ^ℓ in ℓ^{th} layer of TDN by SGD (10).
    Update $\boldsymbol{\beta}^\ell$ in ℓ^{th} layer of TDN by QP (15).
Return deep representation $\mathbf{R} \leftarrow \mathbf{Z}^{l-1}$ and classifier h.

With the closed-form expectations, we can compute mTDA without explicit corruption and achieve a faster algorithm.

4.2 Learning Kernel Parameters

Theoretically, the optimal kernel parameter $\boldsymbol{\beta}$ for MK-MMD can be learned by jointly maximizing the two-sample testing power and minimizing the Type II error of degenerated two-sample test cases $d_k^2(p,q) \to 0$ when $p \neq q$ [12]. However, such an optimization problem is not intuitive. We opt to compute an approximation of the Type II error by choosing an optimal multi-kernel k that maximizes the MK-MMD [46]

$$\max_{\mathbf{1}^\mathsf{T}\boldsymbol{\beta}^\ell=1, \boldsymbol{\beta}^\ell \geq \mathbf{0}} \left\| \mathbb{E}_p \phi(\mathbf{z}^{s,\ell}) - \mathbb{E}_q \phi(\mathbf{z}^{t,\ell}) \right\|_{\mathcal{H}_k}^2 - \varepsilon \left\| \boldsymbol{\beta}^\ell \right\|_2^2, \quad (14)$$

where $\varepsilon = 10^{-3}$ is a small penalty to bound the magnitude of $\boldsymbol{\beta}^\ell$. Denote by $\mathbf{d} = (d_1, d_2, \ldots, d_m)^\mathsf{T}$ the MMDs, where each d_u is the MMD computed using a base Gaussian kernel k_u by the B-test introduced in Equation (9). Problem (14) can be reduced to a constrained Quadratic Program (QP) as

$$\min_{\mathbf{1}^\mathsf{T}\boldsymbol{\beta}^\ell=1, \boldsymbol{\beta}^\ell \geq \mathbf{0}} \varepsilon \boldsymbol{\beta}^{\ell\mathsf{T}} \boldsymbol{\beta}^\ell - \mathbf{d}^\mathsf{T} \boldsymbol{\beta}^\ell, \quad (15)$$

which can be solved efficiently using standard QP packages. The complete procedures for learning both TDA and TDN models are summarized in Algorithm 1. All steps in our algorithm scale linearly to both feature dimension and sample size, hence the overall computational complexity is $O(nd)$.

4.3 Generalization Error Analysis

We analyze the expected target-domain risk of TDN, making use of the theory of domain adaptation [47], [18], [19] and kernel embedding of probability distributions [48], [4], [12].

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

$$\epsilon_t(h) \leqslant \epsilon_s(h) + 2d_k(p,q) + C, \quad (16)$$

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

Proof sketch: The theoretical result in Ben-David et al. [47] shows that $\epsilon_t(h) \leqslant \epsilon_s(h) + d_{\mathcal{H}}(p,q) + C_0$, where $d_{\mathcal{H}}(p,q)$ is the $\mathcal{H}$-divergence that characterizes the discrepancy between distributions p and q by a rich hypothesis space $\mathcal{H}$,

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

The $\mathcal{H}$-divergence relies on the capacity of the hypothesis space $\mathcal{H}$ to distinguish distributions p from q, and $\eta \in \mathcal{H}$ can be viewed as a *two-sample* classifier. By choosing η as a Parzen window classifier [48], $d_{\mathcal{H}}(p,q)$ can be bounded by the risk of the Parzen window classifier (Equation (18), Line 2), which is equivalent to the MK-MMD as revealed by [48]:

$$\begin{aligned} d_{\mathcal{H}}(p,q) &\leqslant \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + C_1 \\ &\leqslant 2 \left(1 - \inf_{\eta \in \mathcal{H}} \left[\sum_{i=1}^{n_s} \frac{L_{+1}[\eta(\mathbf{x}_i^s)]}{n_s} + \sum_{j=1}^{n_t} \frac{L_{-1}[\eta(\mathbf{x}_j^t)]}{n_t} \right] \right) + C_1 \\ &= 2\left(1 + d_k(p,q)\right) + C_1, \end{aligned} \quad (18)$$

where $L(\cdot)$ is the linear loss function for the Parzen window classifier η, and $L_{+1}[\eta] \triangleq -\eta, L_{-1}[\eta] \triangleq \eta$. By explicitly minimizing MK-MMD in multiple layers of TDN, the representation learned by the proposed TDN model can decrease the upper bound on target risk. The source classifier and the two-sample classifier together provide a way to assess the adaptation performance, and can facilitate model selection. Note that we maximize MK-MMD w.r.t. $\boldsymbol{\beta}$ (14) to minimize Type II test error, and to help the Parzen window classifier achieve minimal risk of two-sample discrimination in (18).

5 Experiments

We perform a comprehensive experimental study on diverse real-life domain adaptation problems to evaluate both the effectiveness and scalability of the proposed TDA and TDN models, including sentiment polarity prediction, email spam filtering, newsgroup content classification, and visual object recognition, with a specific focus on the respective effects of unsupervised pre-training, supervised fine-tuning, and optimal two-sample matching. Datasets, codes and configurations used in the evaluation will be made available online.

5.1 Datasets

5.1.1 Multi-Domain Sentiment Dataset

The Multi-Domain Sentiment Dataset[1] [2] has been widely adopted as the benchmark dataset for domain adaptation and sentiment analysis. It contains a collection of product reviews from Amazon.com about four product domains: *books* (**B**), *dvds* (**D**), *electronics* (**E**), and *kitchen appliances* (**K**). Each review is assigned with a positive polarity (higher than 3 stars) or with a negative polarity (3 stars or lower)

1. http://www.cs.jhu.edu/~mdredze/datasets/sentiment

and is represented by the term frequency (TF). Each domain consists of 2,000 labeled reviews and approximately 4,000 unlabeled ones (varying slightly between domains) and the two classes are exactly balanced. Most previous methods [2], [3], [27], [15] provide results on this dataset based on 12 sentiment transfer tasks: **D**→**B**, **E**→**B**, **K**→**B**, **B**→**D**, **E**→**D**, **K**→**D**, **B**→**E**, **D**→**E**, **K**→**E**, **B**→**K**, **D**→**K**, **E**→**K**, where the notation before arrow corresponds to the source domain and the notation after arrow corresponds to the target domain. Detailed statistics of this dataset are summarized in Table 2.

5.1.2 Email Spam Filtering Dataset

The email spam filtering dataset released by ECML/PKDD 2006 Discovery Challenge[2] (Task A) contains 4 separate user inboxes, which can be grouped into *private* inboxes **u1**, **u2**, **u3**, and *public* inbox **u**∗. Each private inbox consists of 1,250 spam and 1,250 non-spam emails of private users, and the public inbox consists of 2,000 spam and 2,000 non-spam emails from public domain, and each email is represented by the term frequency (TF). The sample distributions are similar within each group but are significantly different between groups. Thus in our experiments, the cross-domain tasks are constructed between different groups of inboxes, which are **u1**→**u**∗, **u2**→**u**∗, **u3**→**u**∗, **u**∗→**u1**, **u**∗→**u2**, and **u**∗→**u3**. For example, **u1**→**u**∗ denotes the email spam filtering transfer task using **u1** as source domain and **u**∗ as target domain. Detailed statistics of this dataset are summarized in Table 2.

5.1.3 Newsgroup Classification Dataset

The 20-Newsgroups[3] dataset has approximately 20,000 documents distributed evenly in 20 different subcategories. The corpus contains four top categories *comp* (**C**), *rec* (**R**), *sci* (**S**) and *talk* (**T**), each with four subcategories detailed in [24]. We can construct 6 task groups for binary classification by randomly selecting two top categories, resulting in 6 task groups: **C** → **R**, **C** → **S**, **C** → **T**, **R** → **S**, **R** → **T**, **S** → **T**. For each task group **A** → **B**, we randomly select two subcategories from **A** and **B** respectively to form the source domain and the remaining subcategories form the target domain, resulting in $C_4^2 \cdot C_4^2 = 36$ transfer tasks, and in total we can generate $6 \cdot 36 = 216$ transfer tasks. For fair comparison, the 216 newsgroup tasks are constructed using a preprocessed 20-Newsgroups dataset [24], which contains 25,804 features and 15,033 documents, with each document represented by the term frequency-inverse document frequency (TF-IDF).

5.1.4 Visual Object Recognition Dataset

Office-31 [28] is a standard benchmark dataset for domain adaptation in computer vision, which has 4,652 images in 31 categories collected from three distinct domains: *Amazon* (**A**), which contains images downloaded from `Amazon.com`, *Webcam* (**W**) and *DSLR* (**D**), which are images taken by web camera and digital SLR camera in an office with different environmental and photographing variations. *Caltech-256* (**C**) is a standard database for object recognition with 30,607 images and 256 categories. In the expriments, we adopt the **Office-Caltech** dataset[4] [29], which are comprised of the 10 common categories shared by the Office-31 and Caltech-256 datasets and is widely adopted in transfer learning methods. From the dataset we can construct 12 transfer tasks: **A** → **C**, **W** → **C**, **D** → **C**, **C** → **A**, **W** → **A**, **D** → **A**, **C** → **W**, **A** → **W**, **D** → **W**, **C** → **D**, **A** → **D**, **W** → **D**. The dataset is represented with the DeCAF features [49], which are the 4,096-dimensional FC7-layer hidden activations extracted by the deep convolutional neural network (CNN), i.e. AlexNet [42]. Detailed statistics of this dataset are summarized in Table 2.

TABLE 2
Multi-Domain Sentiment, Email, Newsgroups, and Object Datasets

Dataset	Domain	#Docs	#Train	#Test/Val	#Features
Sentiment	Books (**B**)	6,465	2,000	4,465	30,000
	DVD (**D**)	5,586	2,000	3,586	30,000
	Electronics (**E**)	7,681	2,000	5,681	30,000
	Kitchen (**K**)	7,945	2,000	5,945	30,000
Email	Public (**u**∗)	4,000	3,200	800	206,908
	User1 (**u1**)	2,500	2,000	500	206,908
	User2 (**u2**)	2,500	2,000	500	206,908
	User3 (**u3**)	2,500	2,000	500	206,908
Newsgroup	Comp (**C**)	3,870	3,200	670	25,804
	Rec (**R**)	3,968	3,200	768	25,804
	Sci (**S**)	3,945	3,200	745	25,084
	Talk (**T**)	3,250	2,400	850	25,084
Object	Amazon (**A**)	958	800	158	4,096
	Webcam (**W**)	295	200	95	4,096
	DSLR (**D**)	157	100	57	4,096
	Caltech-256 (**C**)	1,123	1,000	123	4,096

5.2 Evaluation Protocols

To fully evaluate the efficacy of the proposed **TDA** and **TDN** models, we compare with a wide range of state of the art domain adaptation and deep learning methods. As baseline, we train a linear **SVM** on the input features of labeled source domain and test it on unlabeled target domain. For cross-domain sentiment classification, Structural Correspondence Learning (**SCL**) [2] and Spectral Feature Alignment (**SFA**) [3] are the most widely applied methods, hence we investigate them on our datasets. We further compare with Co-Training for Domain Adaptation (**CODA**) [27], which has been shown to produce state of the art performance on multi-domain sentiment dataset based on *shallow* transfer learning. Since SCL, SFA, and CODA are specifically tailored to textual domains, we also investigate two classical transfer learning methods, Transfer Component Analysis (**TCA**) [6] and Geodesic Flow Kernel (**GFK**) [29], which are general methods applicable to various domains. In this regards, we investigate latest general-purpose *shallow* methods, Domain Adaptation Machine (**DAM**) [50] and Transfer Kernel Learning (**TKL**) [24], which have created record performance on both the newsgroup dataset and visual object dataset.

Besides the shallow learning methods, we also consider the latest *deep* learning methods for domain adaptation. We choose to compare with the Marginalized Stacked Denoising Autoencoders (**mSDA**) [15], a marginalized variant of the seminal Stacked Denoising Autoencoders (SDA) [13], which was the first success of deep learning methods applied for domain adaptation on the multi-domain sentiment dataset. Note that the proposed TDA and TDN models distinguish clearly from mSDA and SDA by further calibrating distributions across domains using optimal two-sample matching,

2. http://www.ecmlpkdd2006.org/challenge.html
3. http://people.csail.mit.edu/jrennie/20newsgroups
4. http://www-scf.usc.edu/~boqinggo/domainadaptation.html

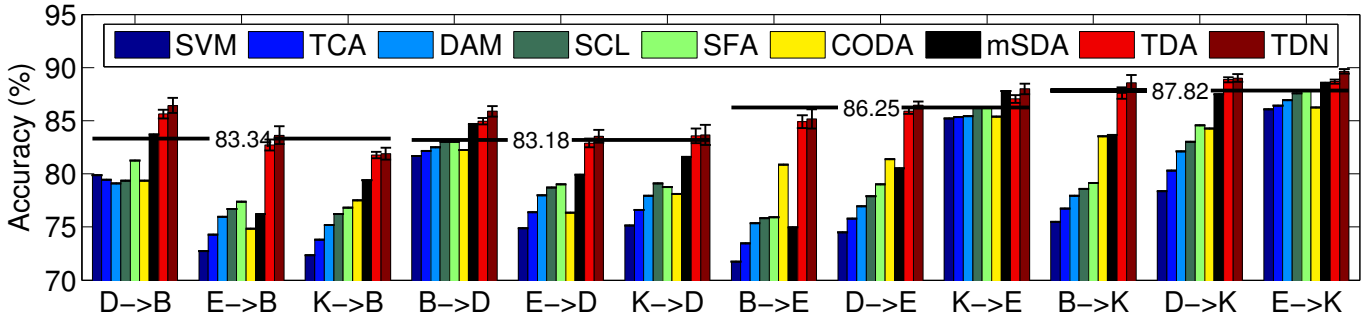


Fig. 2. Accuracy of the 12 transfer tasks on the multi-domain sentiment dataset. All methods are built on the training set of one domain and evaluated on the test sets of the other domain. TDA and TDN outperform all baseline methods on all transfer tasks, with 3.61% accuracy boost over mSDA.

and constitute a unified framework that takes the benefits of both unsupervised pre-training and supervise fine-tuning. Finally, we compare with Deep Adaptation Network (**DAN**) [33], the latest deep transfer learning method that gives the state of the art results on the visual object dataset.

To perform a fair comparative study, we adopt an identical evaluation protocol for all comparison methods [27], [50], [15], [24], [33], and report the average classification accuracy of these methods on all the transfer tasks. For all methods, if there are tunable hyper-parameters, we select their optimal parameters by cross-validation on labeled source data as [6], [15]. For the TDA and TDN models, we also follow standard model selection procedures for deep autoencoders [13] and deep neural networks [42]. More specifically, the number of hidden units, learning rate, corruption probability c, and MK-MMD penalty λ are all selected using cross-validation on labeled source data. To investigate module-wise efficacy of unsupervised pre-training, supervised fine-tuning, and multi-kernel MMD, we evaluate the single-kernel variants of TDA and TDN, respectively termed $\mathbf{TDA_{SK}}$ and $\mathbf{TDN_{SK}}$. Note that, TDA and TDA_{SK} are unsupervised pre-training models that cannot perform classification directly, hence to enable fair comparison, we apply a linear SVM on their last-layer features as [15]. Correspondingly, TDN and TDN_{SK} are supervised fine-tuning models that are initialized by TDA and TDA_{SK} respectively, which can readily take the benefits from the unsupervised pre-training models. The proposed models are implemented using the Pylearn2 open package[5].

5.3 Results and Discussion

We report the experimental results of all methods on applicable transfer tasks and give an in-depth discussion on the respective insights of each method and their interpretations.

5.3.1 Result of Sentiment Classification

The challenge of cross-product sentiment polarity prediction is rooted in the distribution shift, as different words are used to express sentiment in different domains, e.g. words like "blur", "fast", "sharp" are used to comment the *electronics* products, while they carry no opinion in the *books* products. We construct 12 transfer tasks of sentiment classification on this dataset. Figure 2 shows the detailed comparison results of different methods, where each group of bars represents a cross-domain sentiment classification task, and each bar in specific color represents a specific method; the horizontal numbered lines are UpperBound accuracies [15], created by SVM trained with input features on target training data and evaluated on target test data following the train/test splits in Table 2. Note that the target labels are not accessible by all comparison methods during training as there is no labeled target data in unsupervised domain adaptation. For clarity, Table 3 compares the dataset-wise average accuracies.

The proposed TDA and TDN models outperform all the comparison methods on all transfer tasks and significantly boost the performance on 8 out of 12 tasks. The average classification accuracy achieved by TDN on the 12 tasks are **85.99%** and the performance boost is **3.61%** compared to the best baseline method mSDA. It is impressive to observe that TDA and TDN even outperform the UpperBound (denoted by the numbered lines) on 6 tasks and achieve comparable results on the other 6 tasks. This verifies that the proposed models can learn high-quality transferable representation to enable domain adaptation for sentiment classification.

To achieve an in-depth understanding of the proposed models, we further present the results of their variants. (1) The single-kernel variants TDA_{SK} and TDN_{SK} generally achieve better accuracy than the baseline methods, however, they perform fairly worse than the multi-kernel variants TDA and TDN, which highlights that multi-kernel MMD can bridge the domain discrepancy more effectively than single-kernel MMD. The reason is that multiple kernels with different bandwidths can match both low-order moments and high-order moments to minimize the distribution mismatch [12]. (2) TDN, which learns both transferable representation and classifier by optimal two-sample matching, further outperforms TDA. Supervised fine-tuning is critical for domain adaptation since labeled data can be fully exploited to enhance the discriminative power of transferable representation and classifier. Note that TDA serves as an indispensable pre-training for TDN, otherwise TDN cannot be successfully trained due to gradient vanishing [37], [38].

From the results, we also observe that the four domains of the sentiment dataset can be roughly organized into two groups: domains **B** and **D** are similar to each other, as are domains **K** and **E**, but the two groups are very different from each other. Therefore, adapting a supervised classifier from domain **K** to domain **E** is much easier than adapting it from domain **B** or domain **D**. It is interesting to observe

5. https://github.com/lisa-lab/pylearn2

TABLE 3
Average Accuracy (%) of Transfer Tasks on Sentiment (12 Tasks), Email (6 Tasks), Newsgroup (216 Tasks), and Object (12 Tasks) Datasets

Dataset	SVM	TCA	GFK	SCL	SFA	CODA	DAM	TKL	mSDA	DAN	TDA_{SK}	mTDA	TDA	TDN_{SK}	TDN
Sentiment	77.33	78.39	75.12	80.18	80.75	80.84	79.45	76.48	82.38	–	84.91	85.08	85.38	85.74	**85.99**
Email	70.39	67.35	68.97	79.37	78.37	82.41	79.84	68.46	78.68	–	86.19	88.95	89.56	89.80	**90.64**
Newsgroup	82.05	86.31	88.15	84.25	84.86	87.89	87.37	92.41	88.64	–	91.20	91.99	92.64	93.86	**94.55**
Object	85.45	86.64	86.24	–	–	–	86.94	87.81	86.30	90.55	88.65	87.39	89.74	90.16	**90.61**

that the margin of which TDA and TDN outperform the best comparison method increases with adaptability difficulty. In other words, TDA and TDN perform much better than the comparison methods on the difficult-to-transfer tasks, e.g. **E**→**B**, **B**→**K**. This suggests that for those highly difficult transfer tasks, it is more important to extract high-quality representation that manifests domain-invariant structures. The proposed models establish this goal by taking all advantages of unsupervised pre-training, supervised fine-tuning, and optimal multi-kernel two-sample matching.

To give a better understanding of the comparison methods, we further discuss their pros and cons. SVM is known to perform fairly well on standard sentiment classification problems, but it may fail when the training data and testing data are sampled from different domains [2]. TCA and GFK are generic domain adaptation methods, which correct the distribution mismatch using shallow learning architectures (PCA in their cases) but they cannot outperform SVM much, at least for the sentiment tasks if not the case elsewhere. In particular, TCA uses single-kernel MMD to match different distributions, and is less effective than TDA based on MK-MMD. SCL and SFA have been widely recognized as the state of the art domain adaptation methods specifically designed for sentiment polarity prediction. Their effectiveness stems from exploring the domain knowledge of natural languages, that is, identifying domain-shared words as *pivot* features to construct domain-invariant feature subspace for adaptation. However, it still remains unclear how to identify a good set of pivot features for these two methods. CODA improves SCL and SFA by adapting co-training for iterative sample- and feature-selection of TF-IDF features, based on their relevance to the source and target domains. In general, SCL, SFA, and CODA may be limited to the text domains. While TKL shows state of the art results on text and image classification problems, it gives degenerated performance for the sentiment tasks due to the violation of its power-law distribution assumption on kernel eigenspectrum [24]. Thus TKL may only work well for some restricted scenarios. As shallow architectures cannot create compact representation to capture abstract domain-invariant knowledge structure [20], the shallow domain adaptation methods have been surpassed by standard deep learning methods SDA and mSDA which do not explicitly consider the domain mismatch [13], [15]. The proposed TDA and TDN models significantly outperform mSDA by correcting the domain mismatch, which highlights the importance of integrating both deep learning and optimal two-sample matching for domain adaptation.

5.3.2 Result of Email Spam Filtering

The difficulty of cross-inbox email spam filtering is caused by the distribution shift, since different users usually have personalized rules or preferences for detecting email spams. We construct 6 email spam filtering tasks by transferring across public and private groups of inboxes. Note that our tasks are much more difficult than those in [7], which performs transfer within the group of private inboxes **u1**∼**u3**. This can be seen by comparing the base SVM performance: SVM achieved 70.39% on our tasks while it reported 94.9% on those tasks [7]. Figure 3 demonstrates the detailed classification accuracy of all the applicable methods and Table 3 summarizes their average classification accuracy. The proposed TDN model outperforms the best baseline CODA by very large margin of **8.23%**. It is very important to observe that, standard deep learning method mSDA performs much worse than shallow domain adaptation method CODA on this dataset. This reveals that only extracting deep representation without explicitly matching domain distributions may not be good enough for robust domain adaptation, especially when the domain discrepancy is substantially large. Another observation is that CODA consistently outperforms SCL and SFA on this dataset, which further highlights the difficulty in identifying a good set of pivot features for SCL and SFA that behave similarly across domains. Meanwhile, TDA and TDN are generic learning methods that do not need such domain-specific heuristic to engineer the features.

From the results, we also observe an interesting *asymmetry* property of domain adaptation: transferring from task **A** to task **B** is not identical to transferring from task **B** to task **A**. In other words, transferring from public inbox **u**∗ to private inboxes **u1**∼**u3** is much easier than transferring from vice versa. This can be explained as that more generic domains constitute more generic concepts which can be more easily

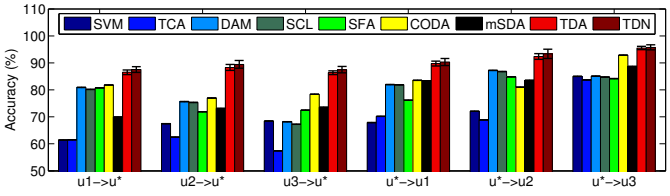


Fig. 3. Accuracy of the 6 transfer tasks on the email spam dataset. TDA and TDN boost the performance by 8.23% over the best baseline CODA.

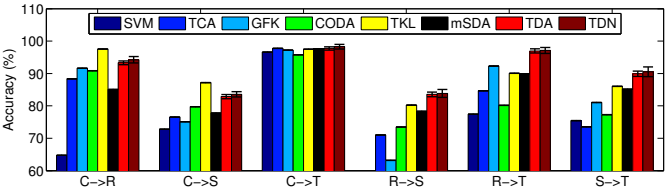


Fig. 4. Accuracy of 6 difficult-to-transfer tasks selected from the newsgroup dataset. TDA and TDN boost the performance by 2.14% over TKL.

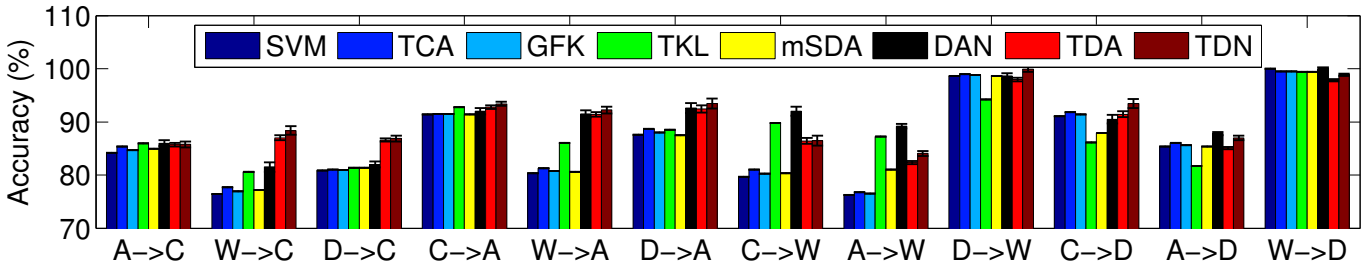


Fig. 5. Accuracy of the 12 transfer tasks on the multi-domain visual object recognition dataset. TDA and TDN achieve comparable performance with the latest state of the art deep convolutional network method DAN while significantly outperform the other methods by an accuracy boost of 2.80%.

adapted to new domains. This suggests that a preferred way for domain adaptation is transferring from generic domains to specific domains, which is consistent with the successful practice of transfer learning from big data such as ImageNet [16]. The results also show that the proposed TDA and TDN models work even better than the baseline methods when transferring from specific domains to more generic domains.

5.3.3 Result of Newsgroup Categorization

The distributions of cross-domain newsgroups are different since the same topic is described by domain-specific words. Newsgroup categorization is very different from sentiment polarity prediction and email spam filtering. Specifically, in sentiment and email tasks, the category of each document can be determined by a few pivot words, which are sentiment words in polarity prediction or spam words in spam filtering. In the newsgroup tasks, however, the category of each document must be determined by considering many content-carrying words, hence it is nontrivial to identify a set of pivot features for newsgroup content classification. Table 3 summarize the average accuracy of all applicable methods on all the 216 newsgroup transfer tasks, while for detailed comparison, Figure 4 also shows the classification accuracies of these methods on the 6 hard-to-transfer tasks. In particular, TKL has been shown to give the state of the art performance on these content-based transfer tasks, although it performs worse on the sentiment and email tasks. Again, TDA and TDN significantly outperform the baseline methods on most tasks, which highlights the advantages of TDA and TDN to be applicable to general-purpose transfer tasks.

5.3.4 Result of Object Recognition

Object recognition is an important computer perception task for multimedia data mining, which is very different from text mining problems. The distribution shift across visual domains is often caused by substantial variations in pose, occlusion, or illumination. Many domain-specific methods including SCL, SFA, and CODA cannot be applied to visual domains. On the other hand, the deep convolutional neural network (CNN) based methods have created breakthroughs in computer vision fields [42], [16]. In particular as shown in Figure 5, the DAN model proposed by Long et al. [33] has given the latest state of the art results for object recognition tasks, but it is a domain-specific method not applicable to text domains. It is promising to see that the proposed TDN model can achieve comparable performance as DAN while significantly outperforms all the other comparison methods. A defense for DAN is that TDN is based on DeCAF features [49], which are also extracted using the deep convolutional neural networks [42]. Nonetheless, DeCAF is only used as a feature extractor for TDN. Decoupling feature extraction from model learning makes TDN a general transfer learner.

5.4 Empirical Analysis

We will go deeper into TDA and TDN by investigating the deep consolidation, domain discrepancy reduction, parameter sensitivity, visualization analysis and scalability study.

5.4.1 Deep Consolidation

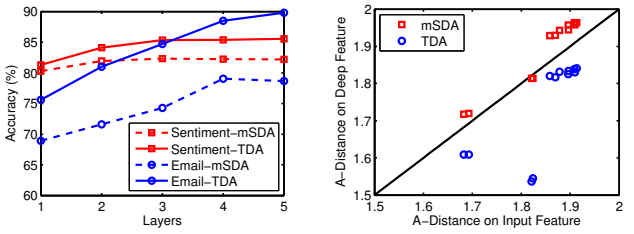
An advantage of deep learning is the capability to extract hierarchical representation. Thus we are curious to find out whether the transferability of deep representation can be enhanced with more layers in the TDA and TDN models. Figure 6(a) demonstrates the average classification accuracy of all transfer tasks on the sentiment and email datasets respectively, with varying number of layers increased from 1 to 5. As expected, both mSDA and TDA perform increasingly better with more layers, which extract more abstract feature representations. This verifies that deep architectures can substantially enhance the representation transferability across domains to enable effective domain adaptation.

It is interesting to observe that the margin of which TDA outperforms mSDA also increases by using more layers. In other words, TDA can outperform mSDA by an even larger margin when more layers of autoencoders are stacked. This can be explained by the deep consolidation of representation invariance. With more layers stacked, the extracted feature representation will be more compact and nonlinear. This naturally enhances the adaptation capability of MK-MMD, since MK-MMD relies heavily on appropriate nonlinear representation to reduce the domain distribution discrepancy.

5.4.2 Proxy-A-Distance

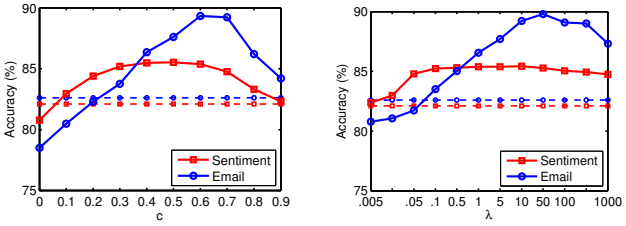
The theoretical results in Ben-David et al. [18] suggested the Proxy-A-Distance (PAD) as a measure of similarity between two probability distributions. They showed that the PAD between the source and target distributions is a crucial part of a generalization error bound for domain adaptation. They hypothesized that it should be difficult to discriminate between the source and target domains if an effective knowledge transfer is established between them, since this would imply similar feature distributions. In practice, computing exact PAD is intractacle and one has to compute a proxy. The approximate PAD is defined as $\hat{d}_A = 2\,(1 - 2\epsilon)$, where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2554549, IEEE Transactions on Knowledge and Data Engineering



(a) Deep Consolidation (b) Proxy-A-Distance

Fig. 6. Effectiveness verification: (a) deep consolidation of transferable representation, (b) proxy-A-distance for measuring domain discrepancy.



(a) Corruption Probability c (b) Adaptation Penalty λ

Fig. 7. Parameter sensitivity: (a) corruption level c, and (b) adaptation penalty λ. TDA outperforms baselines when $c \in [0.4, 0.8]$, $\lambda \in [0.1, 100]$.

ϵ is the generalization error of a classifier (SVM in our case) trained on the binary classification problem to distinguish input samples between the source and target domains.

Figure 6(b) shows the PAD on input features, mSDA features, and TDA features, respectively. Each point in the figure corresponds to the PAD of one transfer task on the sentiment dataset (12 tasks in total), with the 2-dimensional coordinate defined as $(\hat{d}_A(\text{input}), \hat{d}_A(\text{deep}))$. If the point is on the upper (lower) side of line $y = x$, then the PAD on the deep feature is greater (smaller) than the PAD on the input feature. Figure 6(b) reveals a surprising observation: the PAD increases on the mSDA representation in 11 out of 12 tasks, which implies that distinguishing two domains becomes easier with the mSDA features. This phenomenon is explained in [13], [15] as that the deep representation may disentangle both domain-specific and sentiment-polarity information and learns generally better representation for the input data, which helps both domain discrimination and sentiment prediction. However, following Ben-David et al. [18], the representation of mSDA may deteriorate domain adaptation. We conjecture that the representation extracted by standard deep learning is not sufficiently transferable.

As demonstrated in Figure 6(b), the PAD decreases on TDA representation on all 12 transfer tasks. Based on the theory of Ben-David et al. [18], smaller PAD implies lower generalization error for domain adaptation. Hence TDN is guaranteed with better domain generalization performance. From a theoretical perspective, TDA and TDN are superior to prior deep learning based adaptation methods [13], [14], [15], [34]. The superiority of TDA and TDN over TLDA [34] is that there is no theoretical connection between the generalization bound and the KL-divergence adopted in TLDA. Moreover, TDN is also superior to the shallow adaptation methods without deep learning [2], [3], [27], [6], [29], [24], since these methods cannot discover abstract representation.

5.4.3 Parameter Sensitivity

Deep learners based on denoising autoencoders involve an important hyper-parameter, i.e., the corruption probability c. In addition, the adaptation penalty λ is important for the proposed TDA and TDN models. Although these hyper-parameters can be selected using cross-valuation, insensitive parameter performance is desirable for real-life scenarios. Therefore, we conduct sensitivity analysis on both the sentiment and email datasets, where the average classification accuracy of all transfer tasks is computed on each dataset. When testing a specific parameter, we fix other parameters and vary the parameter of our interest. For instance, when testing the influence of adaptation penalty parameter λ, we fix the corruption probability $c = 0.6$ for all transfer tasks.

The detailed results are shown in Figures 7(a) and 7(b), with the best results among the comparison methods shown as dashed lines. Both parameters exhibit bell-shaped performance curves, achieving wide ranges of parameter values in which TDA can significantly outperform the best baselines. Generally, TDA works best under moderate probability of feature corruption, i.e. $c \in [0.4, 0.8]$, which is consistent with mSDA. This suggests that the feature corruption should be neither too small that degenerates the deep representation to trivial identity mapping, nor too large that decreases signal-to-noise ratio. Similarly, TDA favors moderate adaptation penalty, i.e. $\lambda \in [0.1, 100]$. This confirms the motivation of joint deep learning and distribution matching, since a good trade-off of them can enhance effective domain adaptation.

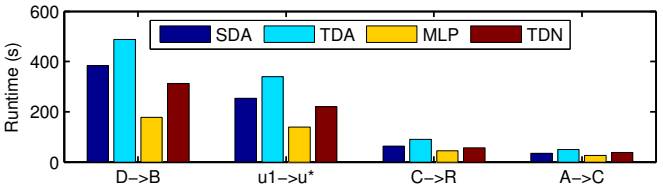


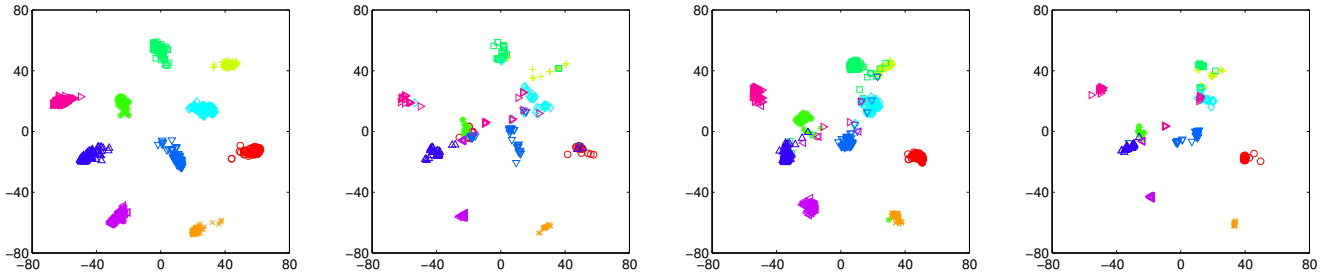
Fig. 8. Scalability analysis: runtimes of SDA vs. TDA and MLP vs. TDN.

5.4.4 Feature Visualization

To demonstrate the transferable representation learned by TDA and TDN, we follow [49] and plot in Figures 9(a)–9(b) and 9(c)–9(d) the t-SNE embeddings of the images in task **A** $\rightarrow$ **W** with DeCAF features and TDN features, respectively. We can observe that with TDN features, the target points are discriminated much better, and the categories between the source and target are aligned much better. Both these observations can explain the superior performance of TDN over DeCAF. Intuitively, TDN can learn both transferable representation and classifier for robust domain adaptation.

5.4.5 Scalability Analysis

We show that the proposed models scale linearly by testing SDA [20] vs. TDA and MLP [21] vs. TDN on several transfer tasks. The runtime in Figure 8 shows that TDA (TDN) has a comparable time complexity as standard SDA (MLP), which are linearly scalable for big domain adaptation applications.



(a) DeCAF Features on Source (b) DeCAF Features on Target (c) TDN Features on Source (d) TDN Features on Target

Fig. 9. Representation visualization: t-SNE of DeCAF features on source (a) and target (b); t-SNE of TDN features on source (c) and target (d).

6 Conclusion and Future Work

In this paper, we have proposed a new domain adaptation framework that jointly learns the transferable representation and classifier to enable scalable domain adaptation, taking the benefits of both deep learning and optimal two-sample matching. Our promising results suggest that it is essential to jointly extract highly abstract feature representation and match different distributions in a unified framework. Also, it is beneficial to explore multiple kernel learning to enhance the transferability of two-sample matching methods. Finally, linear-time algorithms are highly expected for kernel-based domain adaptation methods in the presence of big data.

In the future, we plan to extend our framework to other deep learning methods such as recurrent neural networks, and heterogeneous feature spaces or multiple data sources.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61325008, 61502265), China Postdoctoral Science Foundation (2015T80088), National Science and Technology Supporting Program (2015BAH14F02), and Tsinghua National Laboratory (TNList) Special Fund for Big Data Science and Technology. Jianmin Wang and Mingsheng Long are the corresponding authors.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, 2010.

[2] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Annual Meeting of the Association for Computational Linguistics*, 2007.

[3] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. Int. Conf. World Wide Web*, 2010.

[4] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[5] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Adv. Neural Inf. Process. Syst.*, 2006.

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

[7] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, 2012.

[8] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Int. Conf. Mach. Learn.*, 2013.

[9] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Int. Conf. Comput. Vis.*, 2013.

[10] X. Wang and J. Schneider, "Flexible transfer learning under support and model shift," in *Adv. Neural Inf. Process. Syst.*, 2014.

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

[12] 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.*, 2012.

[13] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Int. Conf. Mach. Learn.*, 2011.

[14] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Int. Conf. Mach. Learn.*, 2011.

[15] M. Chen, Z. E. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Int. Conf. Mach. Learn.*, 2012.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Adv. Neural Inf. Process. Syst.*, 2014.

[17] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Adv. Neural Inf. Process. Syst.*, 2007.

[18] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[19] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Conference on Computational Learning Theory*, 2009.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1." MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[22] W. Zaremba, A. Gretton, and M. Blaschko, "B-test: A non-parametric, low variance kernel two-sample test," in *Adv. Neural Inf. Process. Syst.*, 2013.

[23] K. Zhang, V. W. Zheng, Q. Wang, J. T. Kwok, Q. Yang, and I. Marsic, "Covariate shift in hilbert space: A solution via surrogate kernels," in *Int. Conf. Mach. Learn.*, 2013.

[24] M. Long, J. Wang, J. Sun, and P. S. Yu, "Domain invariant transfer kernel learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, 2015.

[25] P. H. Calais Guerra, A. Veloso, W. Meira Jr., and V. Almeida, "From bias to opinion: a transfer-learning approach to real-time sentiment analysis," in *ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011.

[26] J. Tang, S. Wu, J. Sun, and H. Su, "Cross-domain collaboration recommendation," in *ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012.

[27] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-training for domain adaptation," in *Adv. Neural Inf. Process. Syst.*, 2011.

[28] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. Comput. Vis.*, 2010.

[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*, 2012.
[30] 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.*, 2014.
[31] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," arXiv:1412.3474, Tech. Rep., 2014.
[32] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Mach. Learn.*, 2015.
[33] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Int. Conf. Mach. Learn.*, 2015.
[34] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Int. Joint Conf. Artif. Intell.*, 2015.
[35] 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.
[36] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, 2015.
[37] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[38] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Adv. Neural Inf. Process. Syst.*, 2007.
[39] L. van der Maaten, M. Chen, S. Tyree, and K. Q. Weinberger, "Learning with marginalized corrupted features," in *Int. Conf. Mach. Learn.*, 2013.
[40] M. Chen, K. Weinberger, F. Sha, and Y. Bengio, "Marginalized denoising auto-encoders for nonlinear representations," in *Int. Conf. Mach. Learn.*, 2014.
[41] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *AAAI Conf. on Artif. Intell.*, 2014.
[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012.
[43] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Int. Conf. Mach. Learn.*, 2011.
[44] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, 2009.
[45] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio, "Pylearn2: a machine learning research library," arXiv preprint arXiv:1308.4214, Tech. Rep., 2013.
[46] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, G. R. G. Lanckriet, and B. Schölkopf, "Kernel choice and classifiability for rkhs embeddings of probability distributions," in *Adv. Neural Inf. Process. Syst.*, 2009.
[47] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Adv. Neural Inf. Process. Syst.*, 2007.
[48] 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.*, 2009.
[49] 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.*, 2014.
[50] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw.*, vol. 23, no. 3, pp. 504–518, 2012.

Mingsheng Long graduated from Tsinghua University, China, where he received the B.E. degree in Electrical Engineering and the Ph.D. degree in Computer Science, in 2008 and 2014, respectively. He is a postdoctoral researcher in the School of Software, Tsinghua University. He was a visiting researcher in the AMPLab, UC Berkeley. His research interests include machine learning, data science, and big data systems.

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 has published 100 papers in major journals and conferences, such as *TKDE*, *DMKD*, *WWWJ*, SIGMOD, VLDB, ICDE, SIGIR, ICML, CVPR, and AAAI. He led to develop a product data/lifecycle management system, which has been deployed in hundreds of enterprises in China. He leads to develop a big data management system, LaUDMS.

Yue Cao received the B.E. degree in Computer Software from Tsinghua University, China, in 2014. He is pursuing the Ph.D. degree in Computer Software at Tsinghua University. His research interests include machine learning, data mining, and information retrieval.

Jiaguang Sun received the B.S. degree in Automation Science from Tsinghua University, China, in 1970. He is a full professor at Tsinghua University, where he is the director of the School of Information Science and Technology and the director of the Tsinghua National Laboratory for Information Science and Technology. He is dedicated in teaching and R&D activities in computer graphics, computer-aided design, formal verification of software, and database systems. Prof. Sun has been an academician of the Chinese Academy of Engineering since 1999.

Philip S. Yu received his Ph.D. degree in E.E. from Stanford University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and holds the Wexler Chair in Information Technology. His research interest is on big data, including data mining, data stream, database and privacy. Dr. Yu is a Fellow of the ACM and the IEEE. He is the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data. He was the Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering (2001-2004). He received a Research Contributions Award from IEEE International Conference on Data Mining (2003) and a Technical Achievement Award from IEEE Computer Society (2013).