Deep Transfer Metric Learning

Junlin Hu, Jiwen Lu, *Senior Member, IEEE*, Yap-Peng Tan, *Senior Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*— Conventional metric learning methods usually assume that the training and test samples are captured in similar scenarios so that their distributions are assumed to be the same. This assumption does not hold in many real visual recognition applications, especially when samples are captured across different data sets. In this paper, we propose a new deep transfer metric learning (DTML) method to learn a set of hierarchical nonlinear transformations for cross-domain visual recognition by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain. Specifically, our DTML learns a deep metric network by maximizing the inter-class variations and minimizing the intra-class variations, and minimizing the distribution divergence between the source domain and the target domain at the top layer of the network. To better exploit the discriminative information from the source domain, we further develop a deeply supervised transfer metric learning (DSTML) method by including an additional objective on DTML, where the output of both the hidden layers and the top layer are optimized jointly. To preserve the local manifold of input data points in the metric space, we present two new methods, DTML with autoencoder regularization and DSTML with autoencoder regularization. Experimental results on face verification, person re-identification, and handwritten digit recognition validate the effectiveness of the proposed methods.**

***Index Terms*— Deep metric learning, deep transfer metric learning, transfer learning, face verification, person re-identification.**

I. Introduction

HOW to design a good similarity function plays an important role in many computer vision and pattern recognition tasks. Generally, the optimal similarity function for a given vision problem is task-specific because the underlying data distributions for different tasks are usually different. Recent advances in machine learning have shown that learning a distance metric directly from a set of training examples can usually achieve proposing performance than hand-crafted distance metrics [1]–[3]. In recent years, a variety of metric learning algorithms have been proposed in the literature [2]–[6], and some of them have successfully applied in visual analysis applications such as face recognition [4], [7]–[10], image classification [2], [3], [11], and visual tracking [12].

Existing metric learning methods can be mainly classified into two categories: unsupervised and supervised. For the first category, a low-dimensional subspace or manifold is learned to preserve the geometrical information of the samples. For the second category, a discriminative distance metric is learned to maximize the separability of samples from different classes. Since the label information of training samples is used, supervised metric learning methods are more suitable for the recognition task.

While many supervised metric learning algorithms have been presented in recent years, there are still two shortcomings of these methods: 1) most of them usually seek a single linear distance to transform sample into a linear feature space, so that the nonlinear relationship of samples cannot be well exploited. Even if the kernel trick [13] can be employed to address the nonlinearity issue, these methods still suffer from the scalability problem because they cannot obtain the explicit nonlinear mapping functions; 2) most of them assume that the training and test samples are captured in similar scenarios so that their distributions are assumed to be the same. This assumption doesn't hold in many real visual recognition applications, especially when samples are captured across different datasets.

To this end, in this work, we propose a new deep transfer metric learning (DTML) method for cross-domain visual recognition. Figure 1 illustrates the basic idea of the proposed method. Our method learns a set of hierarchical nonlinear transformations by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain, under which the inter-class variations are maximized and the intra-class variations are minimized, and the distribution divergence between the source domain and the target domain at the top layer of the network is minimized, simultaneously. To better exploit the discriminative information from the source domain, we further develop a deeply supervised transfer metric learning (DSTML) method by including an additional objective on DTML where the output of both the hidden layers and the top layer are optimized jointly. Furthermore, to capture the local manifold structure of input data points as well as easily recover resulting data points in the metric space to their original forms, we employ the autoencoder as a complementary regularizer to regularize

Manuscript received May 10, 2016; revised September 8, 2016; accepted September 16, 2016. Date of publication September 22, 2016; date of current version October 5, 2016. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, in part by the National Natural Science Foundation of China under Grant 61225008, Grant 61672306, Grant 61572271, Grant 61527808, Grant 61373074, and Grant 61373090, in part by the National 1000 Young Talents Plan Program, in part by the National Basic Research Program of China under Grant 2014CB349304, in part by the Ministry of Education of China under Grant 20120002110033, and in part by the Tsinghua University Initiative Scientific Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang-Su Kim. *(Corresponding author: Jiwen Lu.)*

J. Hu and Y.-P. Tan are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jhu007@e.ntu.edu.sg; eyptan@ntu.edu.sg).

J. Lu and J. Zhou are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: lujiwen@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2612827

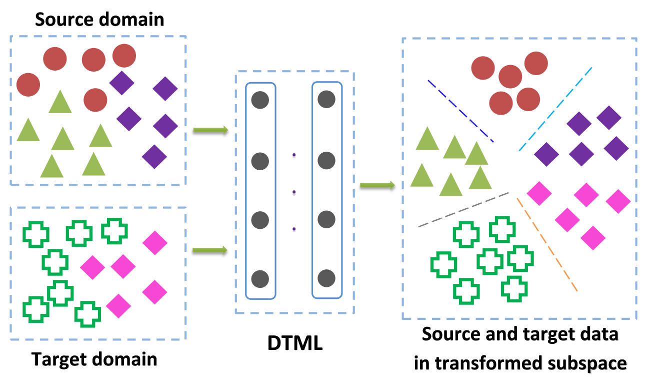


Fig. 1. The basic idea of the DTML method. For each sample in the training sets from the source domain and the target domain, we pass it to the developed deep neural network. We enforce two constraints on the outputs of all training samples at the top of the network: 1) the inter-class variations are maximized and the intra-class variations are minimized, and 2) the distribution divergence between the source domain and the target domain at the top layer of the network is minimized.

our DTML and DSTML methods and further introduce two new methods, deep transfer metric learning with autoencoder regularization (DTML-AE) and deeply supervised transfer metric learning with autoencoder regularization (DSTML-AE), respectively. Experimental results on three cross-domain visual recognition tasks demonstrate the effectiveness of the proposed methods.

This paper is an extension of our conference paper [14]. The new contributions of this paper are summarized as follows:

- We have proposed two new methods, DTML with autoencoder regularization (DTML-AE) and DSTML with autoencoder regularization (DSTML-AE), to preserve the local manifold structure of input data points in the learned metric space. The DTML-AE and DSTML-AE adopt the autoencoder as a complementary regularizer to regularize the DTML and DSTML methods.
- We have conducted more experiments and analysis on three cross-domain visual recognition tasks to show the effectiveness of the proposed methods. Specifically, we include the experiments of the DTML and DSTML methods for cross-domain handwritten digit recognition on two datasets; we include the experimental results of the DTML-AE and DSTML-AE on three cross-domain visual recognition tasks; we include more comparisons with existing state-of-the-art methods; and we also present the parameter sensitivity analysis of our several methods.

The remainder of this paper is organized as follows. Section II briefly introduces the related work. Section III details the proposed DTML and DSTML methods. Section IV presents our DTML-AE and DSTML-AE methods. Section V reports the experimental results and analysis, and Section VI concludes this paper.

II. Related Work

A. Deep Learning

In recent years, deep learning has attracted much attention in computer vision and machine learning due to its superb performance in various tasks. Generally, deep learning aims to learn hierarchical feature representations directly from raw data. Recent advances have shown that deep learning have been successfully applied to many visual tasks such as image classification [15], object detection [16], action recognition [17], and face recognition [18], [19]. Many deep learning models have been proposed in recent years, and representative methods include convolutional neural networks (CNN) [15], [20], deep stacked auto-encoder [17], deep belief networks [21], and deeply-supervised nets [22]. However, most of them aim to learn feature representations via deep model rather than similarity measure. Recently, deep learning has also been used in metric learning, and several metric learning methods have been proposed [4], [12], [23]–[25]. For example, Cai et al. [23] introduced a nonlinear metric learning method using the stacked independent subspace analysis. Hu et al. [4] proposed a discriminative deep metric learning method which employs a conventional neural network by enforcing a large margin criterion at the top layer of the network. While these methods have achieved reasonably good performance, they assume that the training and test samples are captured in the same environments, which is not always satisfied in many real applications. In this work, we presented a deep transfer metric learning approach by learning a deep metric network and considering the distribution difference between source domain and target domain in a unified framework.

B. Transfer Learning

Transfer learning aims to address the problem when the distribution of the training data from the source domain is different from that of the target domain. Over the past decades, a variety of transfer learning algorithms [26]–[29] have been proposed and they can be mainly categorized into two classes: instance-based [30] and feature-based [31]. For the first class, different weights are learned to rank the training samples in the source domain for better learning in the target domain. For the second class, a common feature space is usually learned which can transfer the information learned from the source domain to the target domain. In recent years, several transfer learning techniques have been presented and representative methods include domain transfer support vector machine [32], [33], transfer dimensionality reduction [34], transfer subspace learning [35]–[37], and transfer metric learning [38]–[43]. Concerning the existing transfer metric learning methods, Zha et al. [38] proposed to learn the distance metric by minimizing the divergence between the source metrics and the target metric. Zhang and Yeung [39], [40] introduced a method called transfer metric learning (TML) to learn the distance metric and the task covariances between the source domain and the target domain. Luo et al. [42] proposed a decomposition-based transfer distance metric learning (DTDML) method to seek a sparse combination of the base source metrics to approximate the target metric. In [43], the high-level semantics was used in the transfer metric learning framework. While some proposing results can be obtained by these transfer learning methods, most of them only consider minimizing the distribution difference between the source domain and the target domain by using linear mappings or the kernel trick, which are not effective enough to transfer

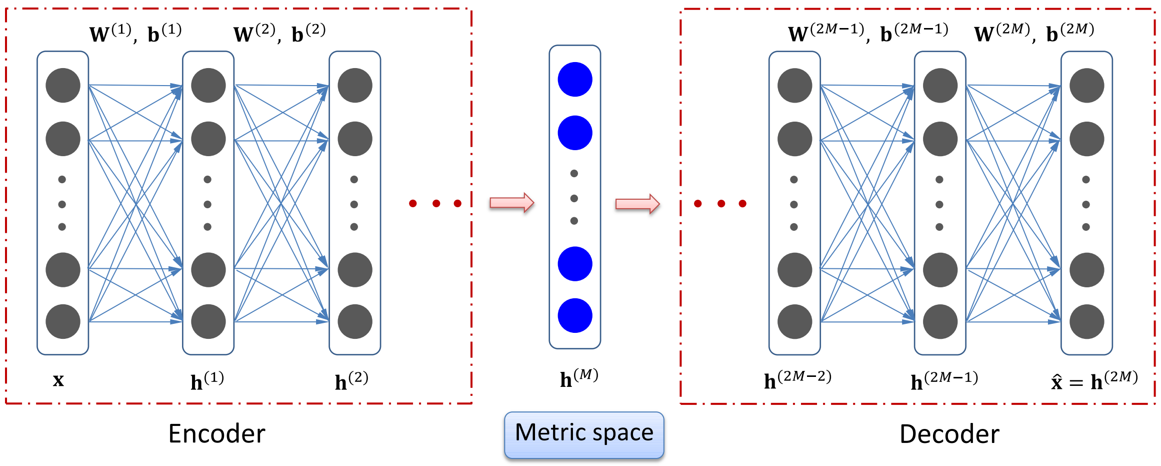

$\mathbf{W}^{(1)}, \mathbf{b}^{(1)}$
$\mathbf{W}^{(2)}, \mathbf{b}^{(2)}$
$\mathbf{W}^{(2M-1)}, \mathbf{b}^{(2M-1)}$
$\mathbf{W}^{(2M)}, \mathbf{b}^{(2M)}$
$\mathbf{x}$
$\mathbf{h}^{(1)}$
$\mathbf{h}^{(2)}$
$\mathbf{h}^{(M)}$
$\mathbf{h}^{(2M-2)}$
$\mathbf{h}^{(2M-1)}$
$\hat{\mathbf{x}} = \mathbf{h}^{(2M)}$
Encoder
Metric space
Decoder

Fig. 2. The network architecture of the autoencoder used in our methods. The $\mathbf{x}$ is an data point in the input space, $\mathbf{h}^{(M)}$ is the resulting representation of the $\mathbf{x}$ in the metric space, and the $\widehat{\mathbf{x}} = \mathbf{h}^{(2M)}$ is the reconstruction of the point $\mathbf{x}$ in the output space.

the knowledge if the distribution difference is large and the transfer functions are usually not explicitly obtained. In this work, we borrow the idea of deep learning and propose a deep transfer metric learning method by learning a discriminative distance network with some information transferred from the source domain.

III. Deep Transfer Metric Learning

In this section, we first introduce the notation used in this work. Then, we present the deep metric learning framework, the proposed deep transfer metric learning method and deeply supervised transfer metric learning method.

A. Notation

Let $\mathcal{X}_s = \{(\mathbf{x}_{si}, y_{si})|i = 1, 2, \ldots, N_s\}$ be the training set in the source domain, which contains N_s examples, where $\mathbf{x}_{si} \in \mathbb{R}^d$ is a d-dimensional feature vector, $y_{si} \in \{1, 2, \ldots, C_s\}$ is the label of $\mathbf{x}_{si}$, and C_s is the number of classes. Similarly, we denote $\mathcal{X}_t = \{\mathbf{x}_{ti}|i = 1, 2, \ldots, N_t\}$ be the training samples in the target domain, where N_t is the number of samples. Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)|i = 1, 2, \ldots, N\}$ be the labeled training set. In our experiments, the labeled training set $\mathcal{X}$ is only sampled from the source domain $\mathcal{X}_s$ (i.e., $\mathcal{X} = \mathcal{X}_s$). Note that the labels y_{si} of training samples in source domain are used to compute the intra-class compactness and the interclass separability.

B. Deep Metric Learning

Unlike most previous metric learning methods which usually seek a single linear distance to transform sample into a linear feature space, we construct a feed-forward neural network to compute the representations of a sample $\mathbf{x}$ by passing it to multiple layers of nonlinear transformations (see [14, Fig. 2]). The key advantage of using such a network to map $\mathbf{x}$ is the nonlinear mapping function can be explicitly obtained. Assume there are $M + 1$ layers in the designed network and $p^{(m)}$ units in the mth layer, where $m = 1, 2, \ldots, M$. The output of $\mathbf{x}$ at the mth layer is computed as:

$$f^{(m)}(\mathbf{x}) = \mathbf{h}^{(m)} = \varphi\left(\mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}\right), \tag{1}$$

where $\mathbf{W}^{(m)} \in \mathbb{R}^{p^{(m)} \times p^{(m-1)}}$ and $\mathbf{b}^{(m)} \in \mathbb{R}^{p^{(m)}}$ are the weight matrix and bias of the parameters in this layer; and φ is a nonlinear activation function which operates component-wisely, such as widely used *tanh* or *sigmoid* functions. The nonlinear mapping $f^{(m)} : \mathbb{R}^d \mapsto \mathbb{R}^{p^{(m)}}$ is a function parameterized by $\{\mathbf{W}^{(i)}\}_{i=1}^m$ and $\{\mathbf{b}^{(i)}\}_{i=1}^m$. For the first layer, we assume $\mathbf{h}^{(0)} = \mathbf{x}$ and $p^{(0)} = d$.

For each pair of samples $\mathbf{x}_i$ and $\mathbf{x}_j$, they can be finally represented as $f^{(m)}(\mathbf{x}_i)$ and $f^{(m)}(\mathbf{x}_j)$ at the mth layer of our designed network, and their distance metric can be measured by computing the squared Euclidean distance between the representations $f^{(m)}(\mathbf{x}_i)$ and $f^{(m)}(\mathbf{x}_j)$ at the mth layer:

$$d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j) = \left\| f^{(m)}(\mathbf{x}_i) - f^{(m)}(\mathbf{x}_j)\right\|_2^2. \tag{2}$$

Following the graph embedding framework, we enforce the marginal fisher analysis criterion [44] on the output of all the training samples at the top layer and formulate a strongly-supervised deep metric learning method as:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha\, S_b^{(M)} + \gamma \sum_{m=1}^{M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2\right), \tag{3}$$

where α ($\alpha > 0$) is a free parameter which balances the importance between intra-class compactness and interclass separability; $\|\mathbf{Z}\|_F$ denotes the Frobenius norm of the matrix $\mathbf{Z}$; γ ($\gamma > 0$) is a tunable positive regularization parameter; $S_c^{(m)}$ and $S_b^{(m)}$ define the intra-class compactness and the interclass separability, which are defined as follows:

$$S_c^{(m)} = \frac{1}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij}\; d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j), \tag{4}$$

$$S_b^{(m)} = \frac{1}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij}\; d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j), \tag{5}$$

where $P_{ij} = 1$ if $\mathbf{x}_j$ is one of k_1-*intraclass* nearest neighbors of $\mathbf{x}_i$, and 0 otherwise; and $Q_{ij} = 1$ if $\mathbf{x}_j$ is one of k_2-*interclass* nearest neighbors of $\mathbf{x}_i$, and 0 otherwise.

C. Deep Transfer Metric Learning

Given target domain data $\mathcal{X}_t$ and source domain data $\mathcal{X}_s$, their probability distributions are usually different in the original feature space when they are captured from different datasets. To reduce the distribution difference, it is desirable to make the probability distribution of the source domain and that of the target domain be as close as possible in the transformed space. To achieve this, we apply the Maximum Mean Discrepancy (MMD) criterion [34] to measure their distribution difference at the mth layer, which is defined as:

$$D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) = \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} f^{(m)}(\mathbf{x}_{ti}) - \frac{1}{N_s} \sum_{i=1}^{N_s} f^{(m)}(\mathbf{x}_{si}) \right\|_2^2. \tag{6}$$

By combining (3) and (6), we formulate DTML as the following optimization problem:

$$\begin{aligned} \min_{f^{(M)}} J &= S_c^{(M)} - \alpha\, S_b^{(M)} + \beta\, D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s) \\ &\quad + \gamma \sum_{m=1}^{M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right), \end{aligned} \tag{7}$$

where β ($\beta \geq 0$) is a regularization parameter.

To solve the optimization problem in (7), we employ the stochastic sub-gradient descent method to obtain the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$. The gradients of the objective function J in (7) with respect to the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are computed as follows:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}^{(m)}} &= \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \left(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \right) \\ &\quad - \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \left(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \right) \\ &\quad + 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} \mathbf{h}_{ti}^{(m-1)^T} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \mathbf{h}_{si}^{(m-1)^T} \Big) \\ &\quad + 2\gamma\ \mathbf{W}^{(m)}, \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{b}^{(m)}} &= \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \right) \\ &\quad - \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \right) \\ &\quad + 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \Big) + 2\gamma\ \mathbf{b}^{(m)}, \end{aligned} \tag{9}$$

where the updating equations are computed as follows:

$$\begin{aligned} \mathbf{L}_{ij}^{(M)} &= \left(\mathbf{h}_i^{(M)} - \mathbf{h}_j^{(M)} \right) \odot \varphi'\left(\mathbf{z}_i^{(M)} \right), \\ \mathbf{L}_{ji}^{(M)} &= \left(\mathbf{h}_j^{(M)} - \mathbf{h}_i^{(M)} \right) \odot \varphi'\left(\mathbf{z}_j^{(M)} \right), \\ \mathbf{L}_{ij}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ij}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_i^{(m)} \right), \\ \mathbf{L}_{ji}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ji}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_j^{(m)} \right), \\ \mathbf{L}_{ti}^{(M)} &= \Big(\frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(M)} - \frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(M)} \Big) \odot \varphi'\left(\mathbf{z}_{ti}^{(M)} \right), \\ \mathbf{L}_{si}^{(M)} &= \Big(\frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(M)} - \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(M)} \Big) \odot \varphi'\left(\mathbf{z}_{si}^{(M)} \right), \\ \mathbf{L}_{ti}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ti}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{ti}^{(m)} \right), \\ \mathbf{L}_{si}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{si}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{si}^{(m)} \right), \end{aligned}$$

where $m = 1, 2, \ldots, M-1$. Here the operation $\odot$ denotes the element-wise multiplication, and $\mathbf{z}_i^{(m)}$ is given as $\mathbf{z}_i^{(m)} = \mathbf{W}^{(m)} \mathbf{h}_i^{(m-1)} + \mathbf{b}^{(m)}$.

Then, $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ can be updated by using the gradient descent algorithm as follows until convergence:

$$\mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \lambda\, \frac{\partial J}{\partial \mathbf{W}^{(m)}}, \tag{10}$$

$$\mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \lambda\, \frac{\partial J}{\partial \mathbf{b}^{(m)}}, \tag{11}$$

where λ is the learning rate.

Algorithm 1 summarizes the detailed optimization procedure of the proposed DTML method.

D. Deeply Supervised Transfer Metric Learning

The objective function of DTML defined in (7) only considers the supervised information of training samples at the top layer of the network, which ignore the discriminative information of the output at the hidden layers. To address this, we further propose a deeply supervised transfer metric learning (DSTML) method to better exploit discriminative information from the output of all layers. We formulate the following optimization problem:

$$\min_{f^{(M)}} J = J^{(M)} + \sum_{m=1}^{M-1} \omega^{(m)}\ h\big(J^{(m)} - \tau^{(m)}\big), \tag{12}$$

where

$$\begin{aligned} J^{(m)} &= S_c^{(m)} - \alpha\, S_b^{(m)} + \beta\, D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) \\ &\quad + \gamma\ \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right), \end{aligned} \tag{13}$$

is the objective function of DTML applied at the mth layer. Here $J^{(M)}$ is the loss of the top layer and $J^{(m)}$ is the loss of the mth *hidden* layer, $m = 1, 2, \ldots, M-1$. The hinge loss function $h(x) = \max(x, 0)$ is used to measure the loss; $\tau^{(m)}$ is a positive threshold, which controls the loss $J^{(m)}$ to show it plays the role in the learning procedure; and $\omega^{(m)}$ balances the importance of the loss obtained at the top layer and the mth hidden layer. The second term in (12) will disappear during

Algorithm 1 DTML

```
Input: Training set: labeled source domain data $\mathcal{X}_s$ and
       unlabeled target domain data $\mathcal{X}_t$; Parameters: $\alpha$,
       $\beta$, $\gamma$, $M$, $k_1$, $k_2$, learning rate $\lambda$, convergence
       error $\varepsilon$, and total iterative number $T$.
for $k = 1, 2, \cdots, T$ do
    Do forward propagation to all data points;
    Compute compactness $S_c^{(M)}$ by (4);
    Compute separability $S_b^{(M)}$ by (5);
    Obtain MMD term $D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s)$ by (6);
    for $m = M, M-1, \cdots, 1$ do
        Compute $\partial J/\partial \mathbf{W}^{(m)}$ and $\partial J/\partial \mathbf{b}^{(m)}$ by
        back-propagation using (8) and (9);
    end
    // Updating weights and biases
    for $m = 1, 2, \cdots, M$ do
        $\mathbf{W}^{(m)} \longleftarrow \mathbf{W}^{(m)} - \lambda \ \partial J/\partial \mathbf{W}^{(m)}$;
        $\mathbf{b}^{(m)} \longleftarrow \mathbf{b}^{(m)} - \lambda \ \partial J/\partial \mathbf{b}^{(m)}$;
    end
    $\lambda \longleftarrow 0.95 \times \lambda$;    // Reducing the learning rate
    Obtain $J_k$ by (7);
    If $|J_k - J_{k-1}| < \varepsilon$, go to Output.
end
Output: Weights and biases $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$.
```

the learning procedure if the overall loss of the mth hidden layer is below the threshold $\tau^{(m)}$.

The gradient of the objective function J in (12) with respect to the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ at the top layer are computed as follows:

$$\frac{\partial J}{\partial \mathbf{W}^{(M)}} = \frac{\partial J^{(M)}}{\partial \mathbf{W}^{(M)}}, \tag{14}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(M)}} = \frac{\partial J^{(M)}}{\partial \mathbf{b}^{(M)}}, \tag{15}$$

For other layers $m = 1, 2, \ldots, M-1$, they are computed as follows:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = \frac{\partial J^{(M)}}{\partial \mathbf{W}^{(m)}} + \sum_{\ell=m}^{M-1} \omega^{(\ell)} \ h'\big(J^{(\ell)} - \tau^{(\ell)}\big) \ \frac{\partial J^{(\ell)}}{\partial \mathbf{W}^{(m)}}, \tag{16}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = \frac{\partial J^{(M)}}{\partial \mathbf{b}^{(m)}} + \sum_{\ell=m}^{M-1} \omega^{(\ell)} \ h'\big(J^{(\ell)} - \tau^{(\ell)}\big) \ \frac{\partial J^{(\ell)}}{\partial \mathbf{b}^{(m)}}, \tag{17}$$

where $h'(x)$ is the derivative of $h(x)$, and we set $h'(x) = 0$ for the non-differentiability point $x = 0$.

For $1 \le m \le \ell \le M$, we have

$$\begin{aligned}
\frac{\partial J^{(\ell)}}{\partial \mathbf{W}^{(m)}} = & \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \left(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \right) \\
& - \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \left(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \right) \\
& + 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} \mathbf{h}_{ti}^{(m-1)^T} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \mathbf{h}_{si}^{(m-1)^T} \Big) \\
& + 2\gamma \ \ \delta(\ell - m) \ \mathbf{W}^{(\ell)},
\end{aligned} \tag{18}$$

$$\begin{aligned}
\frac{\partial J^{(\ell)}}{\partial \mathbf{b}^{(m)}} = & \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \right) \\
& - \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \right) \\
& + 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \Big) \\
& + 2\gamma \ \ \delta(\ell - m) \mathbf{b}^{(\ell)},
\end{aligned} \tag{19}$$

where the delta function $\delta(x) = 0$ holds except $\delta(x) = 1$ at point $x = 0$, and the updating equations for all layers $1 \le m \le \ell - 1$ are computed as follows:

$$\begin{aligned}
\mathbf{L}_{ij}^{(\ell)} &= \left(\mathbf{h}_i^{(\ell)} - \mathbf{h}_j^{(\ell)} \right) \odot \varphi'\left(\mathbf{z}_i^{(\ell)} \right), \\
\mathbf{L}_{ji}^{(\ell)} &= \left(\mathbf{h}_j^{(\ell)} - \mathbf{h}_i^{(\ell)} \right) \odot \varphi'\left(\mathbf{z}_j^{(\ell)} \right), \\
\mathbf{L}_{ij}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ij}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_i^{(m)} \right), \\
\mathbf{L}_{ji}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ji}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_j^{(m)} \right), \\
\mathbf{L}_{ti}^{(\ell)} &= \Big(\frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(\ell)} - \frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(\ell)} \Big) \odot \varphi'\left(\mathbf{z}_{ti}^{(\ell)} \right), \\
\mathbf{L}_{si}^{(\ell)} &= \Big(\frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(\ell)} - \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(\ell)} \Big) \odot \varphi'\left(\mathbf{z}_{si}^{(\ell)} \right), \\
\mathbf{L}_{ti}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ti}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{ti}^{(m)} \right), \\
\mathbf{L}_{si}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \mathbf{L}_{si}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{si}^{(m)} \right).
\end{aligned}$$

We iteratively update $\{\mathbf{W}^{(m)}\}_{m=1}^{M}$ and $\{\mathbf{b}^{(m)}\}_{m=1}^{M}$ using the gradient descent method until the DSTML reaches a certain terminating condition.

IV. Deep Transfer Metric Learning With Autoencoder Regularization

Recent advances in the representation learning and deep learning have shown that the autoencoder and its variants enjoy some good properties in preserving locality of data and recovering original data points from their resulting representations [45]. Inspired by this, in the learned metric space, to capture the local manifold structure of input data points as well as easily recover resulting data points in the metric space to their original forms, we adopt the autoencoder as a complementary regularizer to the DTML and DSTML methods and further introduce two new methods: deep transfer metric learning with autoencoder regularization (DTML-AE) and deeply supervised transfer metric learning with autoencoder regularization (DSTML-AE), respectively. Figure 2 shows the basic framework of the autoencoder used in our DTML-AE and DSTML-AE methods, which mainly consists of two parts, encoder and decoder.

A. DTML With Autoencoder Regularization

Given a data point $\mathbf{x} \in \mathbb{R}^{p^{(0)}}$ in the input space, the autoencoder first encodes it through the encoding part (i.e., first M layers) to obtain the resulting representation $\mathbf{h}^{(M)}$ in the metric space, and then decodes $\mathbf{h}^{(M)}$ through the decoding part (i.e., last M layers) to obtain the reconstruction $\widehat{\mathbf{x}} = \mathbf{h}^{(2M)}$ of the $\mathbf{x}$ in the output space, i.e.,

$$\mathbf{h}^{(M)} = f^{(M)}(\mathbf{x}) \in \mathbb{R}^{p^{(M)}}, \tag{20}$$

$$\widehat{\mathbf{x}} = \mathbf{h}^{(2M)} = f^{(2M)}(\mathbf{x}) \in \mathbb{R}^{p^{(2M)}}, \tag{21}$$

in which the nonlinear mapping $f^{(2M)} : \mathbb{R}^{p^{(0)}} \mapsto \mathbb{R}^{p^{(2M)}}$ is a function parameterized by $\{\mathbf{W}^{(m)} \in \mathbb{R}^{p^{(m)} \times p^{(m-1)}}\}_{m=1}^{2M}$ and $\{\mathbf{b}^{(m)} \in \mathbb{R}^{p^{(m)}}\}_{m=1}^{2M}$, and we have constraints $p^{(m)} = p^{(2M-m)}$, $m = 1, 2, \ldots, 2M$. These constraints are used to make sure that encoder and decoder networks are mirror symmetry for simplicity. In autoencoder term, we just minimize the reconstruction error between output layer ($m = 2M$) and input layer ($m = 0$). By minimizing the reconstruction error $\| f^{(2M)}(\mathbf{x}) - \mathbf{x}\|_2^2$ of each data point of the training set, we can obtain the parameters of this autoencoder.

Then we exploit the autoencoder as a complementary regularizer to the DTML and formulate the proposed DTML-AE method as following optimization problem:

$$\begin{aligned}\min_{f^{(M)},\ f^{(2M)}} J &= S_c^{(M)} - \alpha\ S_b^{(M)} + \beta\ D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s) \\ &\quad + \gamma \sum_{m=1}^{2M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right) \\ &\quad + \frac{\theta_t}{N_t} \sum_{i=1}^{N_t} \| f^{(2M)}(\mathbf{x}_{ti}) - \mathbf{x}_{ti} \|_2^2 \\ &\quad + \frac{\theta_s}{N_s} \sum_{i=1}^{N_s} \| f^{(2M)}(\mathbf{x}_{si}) - \mathbf{x}_{si} \|_2^2 \\ &= J_{\mathrm{DTML}} + J_{\mathrm{AE}}, \end{aligned} \tag{22}$$

where J_{DTML} is the objective function of the DTML in (7), and J_{AE} is the autoencoder regularization of the source domain and target domain data, i.e.,

$$\begin{aligned} J_{\mathrm{AE}} &= \frac{\theta_t}{N_t} \sum_{i=1}^{N_t} \| f^{(2M)}(\mathbf{x}_{ti}) - \mathbf{x}_{ti} \|_2^2 \\ &\quad + \frac{\theta_s}{N_s} \sum_{i=1}^{N_s} \| f^{(2M)}(\mathbf{x}_{si}) - \mathbf{x}_{si} \|_2^2 \\ &\quad + \gamma \sum_{m=M+1}^{2M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right), \end{aligned} \tag{23}$$

in which θ_t and θ_s are two positive regularization parameters to balance the importance of the autoencoder regularization to the whole objective function.

Next the gradient descent based method is applied to update the parameters $\{\mathbf{W}^{(m)}\}_{m=1}^{2M}$ and $\{\mathbf{b}^{(m)}\}_{m=1}^{2M}$. The partial derivative of the objective function J in (22) with regard to $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, $m = 1, 2, \ldots, 2M$, are presented as:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = u(M - m)\ \frac{\partial J_{\mathrm{DTML}}}{\partial \mathbf{W}^{(m)}} + \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{W}^{(m)}}, \tag{24}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = u(M - m)\ \frac{\partial J_{\mathrm{DTML}}}{\partial \mathbf{b}^{(m)}} + \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{b}^{(m)}}, \tag{25}$$

where $u(M - m)$ is the unit step function of the variable m, i.e., it holds that $u(M - m) = 1$ for $1 \le m \le M$, and $u(M - m) = 0$ for $M < m \le 2M$. The $\partial J_{\mathrm{DTML}}/\partial \mathbf{W}^{(m)}$ and $\partial J_{\mathrm{DTML}}/\partial \mathbf{b}^{(m)}$ can be calculated by Eq. (8) and Eq. (9), $m = 1, 2, \ldots, M$. Moreover, the $\partial J_{\mathrm{AE}}/\partial \mathbf{W}^{(m)}$ and $\partial J_{\mathrm{AE}}/\partial \mathbf{b}^{(m)}$ for $m = 1, 2, \ldots, 2M$ are given as follows:

$$\begin{aligned} \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{W}^{(m)}} &= \frac{2\theta_t}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{\mathrm{AE},ti}^{(m)} \mathbf{h}_{ti}^{(m-1)T} + \frac{2\theta_s}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{\mathrm{AE},si}^{(m)} \mathbf{h}_{si}^{(m-1)T} \\ &\quad + 2\gamma\ u(m - M - 1)\ \mathbf{W}^{(m)}, \end{aligned} \tag{26}$$

$$\begin{aligned} \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{b}^{(m)}} &= \frac{2\theta_t}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{\mathrm{AE},ti}^{(m)} + \frac{2\theta_s}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{\mathrm{AE},si}^{(m)} \\ &\quad + 2\gamma\ u(m - M - 1)\ \mathbf{b}^{(m)}, \end{aligned} \tag{27}$$

in which the $\mathbf{L}_{\mathrm{AE},ti}^{(m)}$ and $\mathbf{L}_{\mathrm{AE},si}^{(m)}$ are computed by

$$\begin{aligned} \mathbf{L}_{\mathrm{AE},ti}^{(2M)} &= \left(f^{(2M)}(\mathbf{x}_{ti}) - \mathbf{x}_{ti} \right) \odot \varphi'\left(\mathbf{z}_{ti}^{(2M)} \right), \\ \mathbf{L}_{\mathrm{AE},si}^{(2M)} &= \left(f^{(2M)}(\mathbf{x}_{si}) - \mathbf{x}_{si} \right) \odot \varphi'\left(\mathbf{z}_{si}^{(2M)} \right), \\ \mathbf{L}_{\mathrm{AE},ti}^{(m)} &= \left(\mathbf{W}^{(m+1)T} \mathbf{L}_{\mathrm{AE},ti}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{ti}^{(m)} \right), \\ \mathbf{L}_{\mathrm{AE},si}^{(m)} &= \left(\mathbf{W}^{(m+1)T} \mathbf{L}_{\mathrm{AE},si}^{(m+1)} \right) \odot \varphi'\left(\mathbf{z}_{si}^{(m)} \right), \end{aligned}$$

for layer $m = 1, 2, \ldots, 2M - 1$; and $\mathbf{z}_{ti}^{(m)}$ is given as $\mathbf{z}_{ti}^{(m)} = \mathbf{W}^{(m)} \mathbf{h}_{ti}^{(m-1)} + \mathbf{b}^{(m)}$.

After having obtained the $\{\mathbf{W}^{(m)}\}_{m=1}^{2M}$ and $\{\mathbf{b}^{(m)}\}_{m=1}^{2M}$, the distance of a pair of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ is measured in the deep metric space by

$$d_{f^{(M)}}^2(\mathbf{x}_i, \mathbf{x}_j) = \left\| f^{(M)}(\mathbf{x}_i) - f^{(M)}(\mathbf{x}_j) \right\|_2^2. \tag{28}$$

And this distance is used for verification and recognition.

B. DSTML With Autoencoder Regularization

Following the similar idea as in DTML-AE, the objective function of the DSTML-AE method can be formulated as:

$$\min_{f^{(M)},\ f^{(2M)}} J = J_{\mathrm{DSTML}} + J_{\mathrm{AE}}, \tag{29}$$

where J_{DSTML} is the objective function of the DSTML in (12); and J_{AE} is the autoencoder regularization given by Eq. (23).

The gradient of the objective function J of the DSTML-AE with respect to $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are calculated by:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = u(M - m)\ \frac{\partial J_{\mathrm{DSTML}}}{\partial \mathbf{W}^{(m)}} + \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{W}^{(m)}}, \tag{30}$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = u(M - m)\ \frac{\partial J_{\mathrm{DSTML}}}{\partial \mathbf{b}^{(m)}} + \frac{\partial J_{\mathrm{AE}}}{\partial \mathbf{b}^{(m)}}, \tag{31}$$

for the $m = 1, 2, \ldots, 2M$, in which $\partial J_{\mathrm{DSTML}}/\partial \mathbf{W}^{(m)}$, $\partial J_{\mathrm{DSTML}}/\partial \mathbf{b}^{(m)}$, $\partial J_{\mathrm{AE}}/\partial \mathbf{W}^{(m)}$, and $\partial J_{\mathrm{AE}}/\partial \mathbf{b}^{(m)}$ are given by (14) – (17), (26), and (27).

V. Experiments

In this section, we evaluate the DTML, DSTML, DTML-AE and DSTML-AE methods on three visual recognition tasks: face verification, person re-identification, and cross-domain handwritten digit recognition. The followings describe the detailed settings and experimental results.

A. Face Verification

Face verification aims to decide whether a given pair of face images are from the same person or not. In this section, we conducted cross-dataset unconstrained face verification where face images were captured in wild conditions so that significant variations such as varying expression, pose, lighting, and background are included in the captured images.

1) Datasets and Experimental Settings: We used the Labeled Faces in the Wild (LFW) [46] and the Wide and Deep Reference (WDRef) [47] datasets for cross-dataset face verification. LFW is a challenging dataset which was developed for studying the problem of unconstrained face verification. There are 13233 face images of 5749 persons in the LFW dataset, which were collected from the internet and 1680 people have two or more images. This dataset was divided into 10 non-overlapping folds and the 10-fold cross validation evaluation was used to test the performance of different face verification methods. For each fold, there are 300 matched (positive) pairs and 300 mismatched (negative) image pairs, respectively.

The WDRef [47] dataset contains 99773 face images of 2995 people, and 2065 subjects have at least 15 images. Since face image captured in WDRef are also from the internet, face images from these two datasets are generally similar. There is no overlapped subject between the LFW and WDRef datasets. In our experiments, we applied a subset which is randomly sampled from WDRef which consists of 15000 images (1500 subjects and each subject has 10 images) to learn the discriminative metric network.

In our settings, we selected the LFW dataset as the target domain data and the WDRef dataset as the source domain data. For each face image in these two datasets, we used the conventional local binary patterns (LBP) feature provided by the authors [47] to represent each face image. Specifically, we first extracted LBP descriptors at five facial landmarks with various scales, and then concatenated them to form a 5900-dimensional feature vector. To further remove the redundancy, each feature vector is reduced to 500 dimension by principal component analysis (PCA) where the projection is learnt on the source domain data. For our proposed DTML and DSTML methods, we designed a deep network with three layers ($M = 2$), and neural nodes from bottom to top layer are set as: $500 \to 400 \to 300$. For the DTML-AE and DSTML-AE, we used an autoencoder of size $500 \to 400 \to 300 \to 400 \to 500$. The *tanh* function was used as the nonlinear activation function in these methods, and the parameters α, β, γ, $\omega^{(1)}$, $\tau^{(1)}$, k_1, k_2, θ_t and θ_s were empirically set as $\alpha = 0.1$, $\beta = 10$, $\gamma = 0.1$, $\omega^{(1)} = 1$, $\tau^{(1)} = 0$, $k_1 = 5$, $k_2 = 10$, $\theta_t = 1$ and $\theta_s = 1$, respectively. The initialized value of the learning rate λ was set as 0.2, and then it gradually reduced by multiplying a factor 0.95 in each iteration. We initialized $\mathbf{W}^{(m)}$ as a matrix with ones on the main diagonal and zeros elsewhere, and set the bias $\mathbf{b}^{(m)}$ as $\mathbf{0}$ for all $m = 1, 2, \ldots, M$.

2) Results and Analysis: We first compared our DTML with the shallow transfer metric learning (STML) method to show the advantage of the deep network. The STML is a special case of our DTML approach where the designed neural network only has two layers (i.e., input layer and output layer) and the linear activation function $\varphi(x) = x$ was used. Table I shows the average verification rate with standard error of the DTML, DSTML, DTML-AE, DSTML-AE and STML. We can observe that DTML and its variants achieve better performance than STML. The reason is that DTML learns several hierarchical nonlinear transformations while STML learns only one linear transformation, so that the nonlinearity can be better exploited for face verification. Moreover, DSTML improves DTML by 1.7% in terms of the mean verification rate. This is because DSTML utilizes the label information from training sampled in the source domain at both the hidden layers and the top layer while DTML only exploit such information at the top layer, such that more discriminative information is exploited in DSTML. We also notice that the DTML-AE and DSTML-AE outperform the DTML and DSTML by 1.5% and 0.9%, respectively. The reason is that DTML-AE and DSTML-AE can preserve the local manifold of data points in the learned metric space to help improve verification accuracy. Table I also lists the comparison of our methods and two existing transfer metric learning methods. The proposed deep transfer metric learning methods consistently perform better than these two transfer metric learning methods at a large gain.

To show the efficacy of knowledge transfer in our methods, we also compared STML, DTML, DSTML, DTML-AE and DSTML-AE with their corresponding metric learning methods without knowledge transfer. To this end, we removed the MMD regularization term in STML, DTML, DSTML, DTML-AE and DSTML-AE, which means the value of the parameter β is set as 0 in these methods. Table II shows

TABLE I

MEAN VERIFICATION RATE WITH STANDARD ERROR (%) OF DIFFERENT TRANSFER METRIC LEARNING METHODS ON THE LFW DATASET

Method	Accuracy (%)
STML	83.60 ± 0.75
DTML	85.58 ± 0.61
DSTML	87.32 ± 0.67
DTML-AE	87.02 ± 0.47
DSTML-AE	$\mathbf{88.23 \pm 0.45}$
TML [40]	83.25 ± 0.53
DTDML [42]	83.80 ± 0.62

TABLE II

MEAN VERIFICATION RATE WITH STANDARD ERROR (%) OF DIFFERENT METRIC LEARNING METHODS WITH OR WITHOUT KNOWLEDGE TRANSFER ON THE LFW DATASET

Method	Transfer	Accuracy (%)
LMNN [3]	*yes*	79.13 ± 0.58
LMNN [3]	*no*	77.27 ± 0.52
DDML [4]	*yes*	84.63 ± 0.71
DDML [4]	*no*	83.16 ± 0.80
STML	*yes*	83.60 ± 0.75
STML ($\beta = 0$)	*no*	82.57 ± 0.81
DTML	*yes*	85.58 ± 0.61
DTML ($\beta = 0$)	*no*	83.80 ± 0.55
DSTML	*yes*	87.32 ± 0.67
DSTML ($\beta = 0$)	*no*	85.25 ± 0.59
DTML-AE	*yes*	87.02 ± 0.47
DTML-AE ($\beta = 0$)	*no*	84.77 ± 0.51
DSTML-AE	*yes*	$\mathbf{88.23 \pm 0.45}$
DSTML-AE ($\beta = 0$)	*no*	86.10 ± 0.58

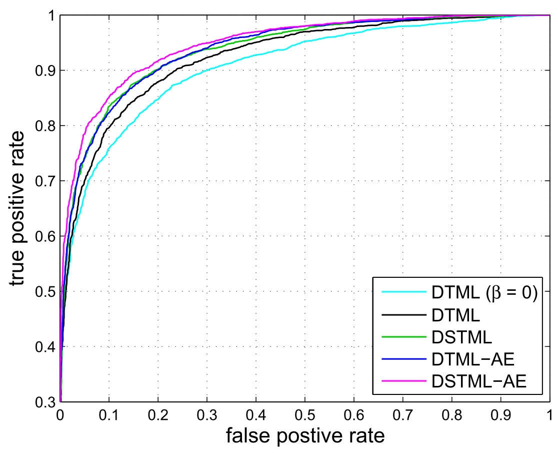


Fig. 3. The ROC curves of several methods on LFW dataset.

the performance comparisons of these methods with and without knowledge transfer from source domain on the LFW dataset. We see that methods with knowledge transfer consistently improve those methods without knowledge transfer by $1\% \sim 2.2\%$ in terms of the mean verification accuracy even though the LFW and WDRef datasets are similar. Figure 3 also shows the ROC curves of several deep metric learning methods, we can observe that transfer metric learning methods outperform those methods without knowledge transfer.

TABLE III

Top r Ranked Matching Accuracy (%) on the VIPeR Dataset With $\#test = 316$ Testing Persons

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	3.99	8.73	12.59	25.32
L_2	-	4.24	8.92	12.66	25.35
DDML [4]	i-LIDS	5.63	12.91	21.71	41.80
	CAVIAR	5.91	13.53	19.86	37.92
	3DPeS	6.67	17.16	23.87	41.65
DTML ($\beta = 0$)	i-LIDS	5.88	13.72	21.03	41.49
	CAVIAR	6.02	13.81	20.33	38.46
	3DPeS	7.20	18.04	25.96	43.80
DTML	i-LIDS	6.68	15.73	23.20	46.42
	CAVIAR	6.17	13.10	19.65	37.78
	3DPeS	8.51	19.40	27.59	47.91
DSTML ($\beta = 0$)	i-LIDS	5.97	15.53	22.70	44.36
	CAVIAR	6.36	16.30	23.52	40.44
	3DPeS	7.95	18.13	25.17	44.90
DSTML	i-LIDS	6.11	16.01	23.51	45.35
	CAVIAR	6.61	16.93	24.40	41.55
	3DPeS	8.58	19.02	26.49	46.77
DTML-AE ($\beta = 0$)	i-LIDS	6.52	15.06	21.97	45.23
	CAVIAR	6.10	14.45	20.92	39.16
	3DPeS	8.39	19.61	27.57	46.85
DTML-AE	i-LIDS	6.86	16.15	24.06	47.93
	CAVIAR	6.45	14.63	21.21	39.54
	3DPeS	8.87	**20.03**	**28.93**	49.30
DSTML-AE ($\beta = 0$)	i-LIDS	6.20	16.11	23.38	45.26
	CAVIAR	6.34	16.72	24.06	41.03
	3DPeS	8.22	18.16	25.95	46.60
DSTML-AE	i-LIDS	6.73	16.52	24.26	47.34
	CAVIAR	6.80	17.15	26.13	43.36
	3DPeS	**8.89**	19.78	28.01	**49.45**

B. Person Re-Identification

Person re-identification aims to recognize person across multiple cameras without overlapping views. This task is very challenging because images of the same subject collected in multiple cameras are usually different due to variations of viewpoint, illumination, pose, resolution and occlusion. While many person re-identification methods have been proposed [48]–[50], there is no much works on cross-dataset person re-identification. In this subsection, we evaluate our deep transfer metric learning methods on cross-dataset person re-identification where only the label information of the source domain is used in the model learning.

1) Datasets and Experimental Settings: The VIPeR dataset [51] consists of 632 intra-personal image pairs captured outdoor by two different camera views, and most of them contain a viewpoint change of about 90 degrees. The i-LIDS dataset [52] contains 476 images of 119 people captured by five cameras at an airport, and each pedestrian has $2 \sim 8$ images. The CAVIAR dataset [53] contains 1220 images of 72 individuals from two different cameras in an indoor shopping mall, with $10 \sim 20$ images per person as well as large variations in resolutions. The 3DPeS dataset [54] has 1011 images of 192 persons collected from 8 different outdoor cameras with significant changes of viewpoint, and most of individuals appear in three different camera views.

In our experiments, all images from these datasets were scaled to 128×48 for feature extraction. For each image, we used two kinds of features descriptor: color and texture histograms. Following the settings in [50], each image was divided into six non-overlapping horizontal stripes. For each stripe, we extracted 16-bins histograms for each color channel of eight channels in the RGB (R, G, B), YUV (Y, U, V) and HSV (H, S) spaces, and computed uniform LBP with 8 neighbors and 16 neighbors respectively. Finally, color and texture histograms from these stripes were concatenated to form a 2580-dimensional feature vector for each image. Then PCA learnt on the source domain data was applied to reduce the dimension of target feature vector into a low dimensional subspace. We also adopted the single-Shot experiment setting [48], [50] to randomly split individuals in each dataset as training set and testing set, and repeated 10 times. For each partition, there were $\#test$ subjects in the testing set, and one image for each person was randomly selected as the gallery set and the remaining images of this person were used as probe images. In DTML and DSTML, we also used a network with three layers ($M = 2$), and its neural nodes are given as: $200 \rightarrow 200 \rightarrow 100$ for all datasets. In DTML-AE and DSTML-AE, we employed an autoencoder of size $200 \rightarrow 200 \rightarrow 100 \rightarrow 200 \rightarrow 200$ for all datasets. The parameter k_1 and k_2 are set as 3 and 10, respectively. Note that if the number of image for a given person is less than 3, all intra-class neighbors were used to compute the intra-class variations. For other parameters, we used the same settings as in face verification experiments on the LFW dataset.

2) Results and Analysis: Tables III – VI show cross-dataset person re-identification performance of our methods on the VIPeR, i-LIDS, CAVIAR and 3DPeS datasets, respectively. The L_1 and L_2 are two baseline methods which directly use L_1 and L_2 norms to compute distance between a probe image and a gallery image in the target domain. And DDML [4] is deep metric learning method which learns a distance metric on

TABLE IV
TOP r RANKED MATCHING ACCURACY (%) ON THE i-LIDS DATASET WITH $\#test = 60$ TESTING PERSONS

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	16.51	28.41	38.28	69.32
L_2	-	16.30	28.25	38.40	69.77
DDML [4]	VIPeR	25.32	45.61	60.27	83.31
	CAVIAR	25.67	45.03	61.38	82.56
	3DPeS	28.71	48.55	62.53	83.15
DTML ($\beta = 0$)	VIPeR	26.27	47.59	62.62	85.07
	CAVIAR	26.15	46.87	62.08	84.78
	3DPeS	30.23	51.60	65.21	85.53
DTML	VIPeR	28.90	51.43	65.47	87.23
	CAVIAR	26.23	49.31	63.99	87.76
	3DPeS	31.01	54.51	65.96	88.66
DSTML ($\beta = 0$)	VIPeR	27.50	49.13	59.92	82.89
	CAVIAR	26.81	47.30	62.96	86.15
	3DPeS	31.44	53.23	66.15	87.29
DSTML	VIPeR	28.35	50.81	61.58	84.72
	CAVIAR	28.37	49.68	64.59	88.68
	3DPeS	33.37	54.56	68.27	89.32
DTML-AE ($\beta = 0$)	VIPeR	27.40	48.87	64.56	86.91
	CAVIAR	27.15	48.00	63.61	86.53
	3DPeS	31.36	52.93	67.02	87.46
DTML-AE	VIPeR	29.43	52.26	66.01	87.85
	CAVIAR	26.70	49.82	64.65	88.39
	3DPeS	31.25	55.10	66.59	89.11
DSTML-AE ($\beta = 0$)	VIPeR	28.10	50.56	61.27	84.45
	CAVIAR	28.06	49.43	64.15	88.23
	3DPeS	32.71	53.62	66.30	88.31
DSTML-AE	VIPeR	28.68	51.27	62.10	85.27
	CAVIAR	28.89	50.10	65.13	89.30
	3DPeS	**34.00**	**55.21**	**68.92**	**90.05**

TABLE V
TOP r RANKED MATCHING ACCURACY (%) ON THE CAVIAR DATASET WITH $\#test = 36$ TESTING PERSONS

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	20.65	36.44	48.52	88.34
L_2	-	20.19	36.43	48.55	87.69
DDML [4]	VIPeR	23.80	42.15	55.61	90.73
	i-LIDS	22.72	41.36	56.92	90.06
	3DPeS	23.85	44.30	57.81	90.27
DTML ($\beta = 0$)	VIPeR	23.71	42.57	56.15	90.55
	i-LIDS	23.09	42.81	58.43	90.41
	3DPeS	25.11	46.71	59.69	91.99
DTML	VIPeR	23.88	42.36	55.60	92.12
	i-LIDS	26.06	47.37	61.70	94.23
	3DPeS	26.10	47.80	61.31	93.02
DSTML ($\beta = 0$)	VIPeR	23.93	42.85	56.76	91.32
	i-LIDS	24.65	43.04	57.30	91.28
	3DPeS	26.70	47.51	62.19	92.34
DSTML	VIPeR	26.05	44.33	57.02	92.80
	i-LIDS	25.91	44.47	58.88	93.33
	3DPeS	28.18	49.96	63.67	94.13
DTML-AE ($\beta = 0$)	VIPeR	24.10	42.95	56.80	91.19
	i-LIDS	23.44	43.20	58.92	91.05
	3DPeS	25.50	47.34	60.53	92.82
DTML-AE	VIPeR	24.73	43.27	57.68	93.03
	i-LIDS	26.41	47.78	62.19	94.50
	3DPeS	26.55	48.01	61.66	93.41
DSTML-AE ($\beta = 0$)	VIPeR	24.18	43.16	57.21	91.97
	i-LIDS	25.10	43.57	57.83	92.19
	3DPeS	27.22	48.10	62.76	93.25
DSTML-AE	VIPeR	26.26	44.69	57.62	93.21
	i-LIDS	26.35	45.23	59.50	93.54
	3DPeS	**28.59**	**50.25**	**63.98**	**94.66**

TABLE VI
TOP r RANKED MATCHING ACCURACY (%) ON THE 3DPeS DATASET WITH $\#test = 95$ TESTING PERSONS

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	26.93	42.21	51.56	69.57
L_2	-	26.95	42.57	51.46	69.84
DDML [4]	VIPeR	29.56	51.03	61.71	78.62
	i-LIDS	27.81	50.29	58.33	77.05
	CAVIAR	30.32	49.36	58.92	79.61
DTML ($\beta = 0$)	VIPeR	30.33	52.18	62.24	82.58
	i-LIDS	29.12	50.07	59.99	78.59
	CAVIAR	31.23	51.88	60.87	81.30
DTML	VIPeR	32.12	54.36	65.92	84.65
	i-LIDS	32.11	52.08	61.63	79.45
	CAVIAR	31.79	51.92	62.78	81.98
DSTML ($\beta = 0$)	VIPeR	30.87	52.63	62.51	82.73
	i-LIDS	31.03	51.85	62.34	83.11
	CAVIAR	31.90	53.00	64.65	82.53
DSTML	VIPeR	32.51	52.97	63.12	83.08
	i-LIDS	31.57	52.54	63.50	84.02
	CAVIAR	32.53	54.29	65.28	84.72
DTML-AE ($\beta = 0$)	VIPeR	30.57	52.51	62.59	83.10
	i-LIDS	29.60	50.83	60.75	78.92
	CAVIAR	31.58	52.25	61.31	82.06
DTML-AE	VIPeR	32.81	**54.90**	**66.76**	85.14
	i-LIDS	32.46	52.82	62.37	80.31
	CAVIAR	32.62	52.76	63.03	82.79
DSTML-AE ($\beta = 0$)	VIPeR	31.05	52.88	62.95	83.27
	i-LIDS	31.20	52.13	62.78	83.81
	CAVIAR	32.18	53.44	65.00	83.56
DSTML-AE	VIPeR	32.82	53.39	63.51	83.70
	i-LIDS	31.99	53.01	64.10	84.53
	CAVIAR	**33.05**	54.75	66.20	**85.69**

the source domain only. From these tables, we can observe that our DTML, DSTML, DTML-AE and DSTML-AE methods achieve better performance than the metric learning method without knowledge transfer in most of cases, which shows that transfer metric learning can help improve the performance of cross-dataset person re-identification. We also notice that DSTML-AE obtains the best performance for most cases, which indicates that exploiting discriminative information from more hidden layers rather than the single output layer is more effective to learn good distance metrics, and employing the autoencoder regularization term can capture the local manifold of data points in the learned metric space.

C. Handwritten Digit Recognition

Handwritten digit recognition is a classic example in machine learning and compute vision. We conduct several experiments on two well-known handwritten digit datasets for cross-domain handwritten digit recognition.

1) Datasets and Experimental Settings: The MNIST [20] handwritten digit dataset totally contains 70000 gray-scale images, in which the size of each image is 28×28 pixels. The USPS handwritten digit dataset consists of 9298 images of size 16×16 pixels. Both MNIST and USPS datasets enjoy ten semantic classes form digit 0 to digit 9. Figure 4 shows several images randomly sampled from the MNIST and USPS datasets, where the images of these two datsets present the difference in appearance and distribution.

In the experiments, each image of the MNIST dataset was scaled to 16×16 pixels. And then each image of two datasets was represented by a 256-dimensional feature vector of raw pixels. Following the similar evaluation setting as in [55], we randomly chose 100 labeled images per class from the source domain (Note that these 1000 labeled images were also used

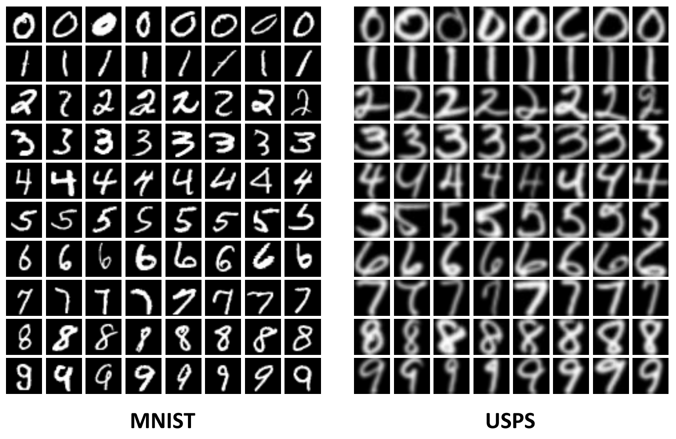


Fig. 4. Handwritten digit images are from MNIST and USPS datasets.

as the gallery set in test.) and 1000 unlabeled images from the target domain to form the training set. In test, 10 images per category were sampled from the remaining samples of the target domain to form the probe set. We repeated this split 5 times and reported the average recognition accuracies for performance evaluation. Regarding the parameter settings, we adopted neural nodes of size 256 → 200 → 100 for the DTML and DSTML methods, and employed the autoencoder of size 256 → 200 → 100 → 200 → 256 for the DTML-AE and DSTML-AE methods. For other parameters, we followed the same settings as in face verification experiments.

2) Results and Analysis: The average recognition accuracy of our transfer metric learning methods, including STML, DTML, DSTML, DTML-AE and DSTML-AE on the two cross-domain settings (i.e., MNIST → USPS, and USPS → MNIST) are illustrated in Table VII. From these results, we can notice that all transfer metric learning methods consistently outperform their partners which only learn the distance metrics on the source domain. This further shows effectiveness of our proposed approaches on exploiting information of source domain to help cross-domain recognition. We also notice that 1) DSTML and DSTML-AE show better accuracies than DTML and DTML-AE; and 2) DTML-AE and DSTML-AE consistently outperform DTML and DSTML methods. The reason is that DSTML and DSTML-AE can make use of the label information on the hidden layers and the output layer to guide parameters learning, and DTML-AE and DSTML-AE can preserve the local manifold of data points in the learned metric space, respectively. In addition, Table VII lists performance comparison of our methods and several state-of-the-art transfer learning methods, including SGF [56], GFK [57], LSDT [55] and NLSDT [55]. We can find that the DSTML-AE obtains the best accuracy, and our DTML-AE is comparable to NLSDT [55] on two cross-domain settings.

TABLE VII

THE AVERAGE RECOGNITION ACCURACY (%) OF SEVERAL METHODS ON MNIST AND USPS DATASETS. NOTE THAT THE ARROW "→" DENOTES THE DIRECTION FROM SOURCE DOMAIN TO TARGET DOMAIN

Method	MNIST → USPS	USPS → MNIST
STML	80.5 ± 0.6	72.8 ± 0.9
STML ($\beta = 0$)	71.3 ± 0.7	61.6 ± 1.3
DTML	85.1 ± 0.5	76.2 ± 0.8
DTML ($\beta = 0$)	73.9 ± 0.8	65.3 ± 1.1
DSTML	86.5 ± 0.6	78.5 ± 0.7
DSTML ($\beta = 0$)	74.4 ± 0.9	66.6 ± 1.2
DTML-AE	87.2 ± 0.7	78.7 ± 0.8
DTML-AE ($\beta = 0$)	74.6 ± 0.8	67.3 ± 1.0
DSTML-AE	$\mathbf{88.7 \pm 0.4}$	$\mathbf{79.3 \pm 0.6}$
DSTML-AE ($\beta = 0$)	75.8 ± 0.5	68.5 ± 1.1
SGF [56]	79.2 ± 0.9	71.1 ± 0.7
GFK [57]	82.6 ± 0.8	74.9 ± 0.9
LSDT [55]	79.3 ± 0.8	70.5 ± 1.4
NLSDT [55]	87.4 ± 0.5	79.1 ± 0.8

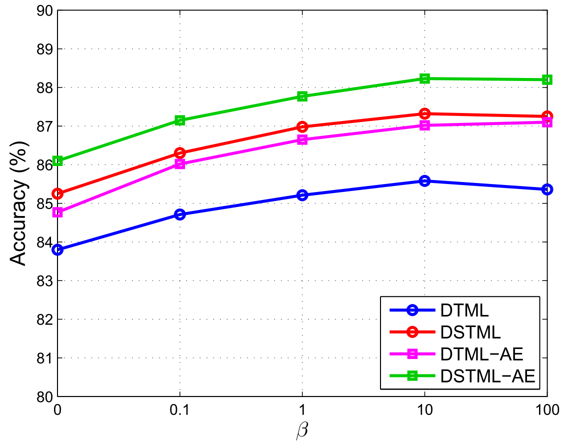


Fig. 5. The mean accuracy of the DTML, DSTML, DTML-AE and DSTML-AE methods with varying values of β on the LFW dataset.

D. Parameter Analysis

In this subsection, we empirically investigate the parameter sensitivity analysis of our methods on LFW and WDRef datasets. While there are several parameters, three of them (β, θ_t and θ_s) are the key parameters and others can be empirically determined by cross validation. Specifically, we fixed α, γ, k_2, τ and ω, and searched β in $\{0, 0.1, 1, 10, 100\}$ and θ_t ($\theta_s = \theta_t$) in $\{0, 0.1, 1, 10, 100\}$ by cross-validation.

1) MMD Regularization β: We first investigated the influence of the MMD regularization parameter β to our transfer metric learning methods. We run DTML, DSTML, DTML-AE and DSTML-AE methods with varying values of β. Specifically, we individually picked β from $\{0, 0.1, 1, 10, 100\}$ while fixing other parameters to their default settings. Figure 5 plots the mean verification accuracy of our four methods under different values of β. From this figure, we can observe that the larger values of β gradually improve the performance of all methods. The reason is that the large β can reduce the distribution difference of source domain and target domain. We also notice that our four methods achieve the relatively stable performance when β falls into $[1, 100]$. Thus, we can easily choose $\beta \in [1, 100]$, and we set $\beta = 10$ in our experiments to obtain stable results for other datasets.

2) Autoencoder Regularization θ_t and θ_s: We then study the sensitivity of the autoencoder regularization parameters θ_t and θ_s for the DTML-AE and DSTML-AE methods. We implemented the DTML-AE and DSTML-AE approaches with various values of θ_t and θ_s in $\{0, 0.1, 1, 10, 100\}$ when

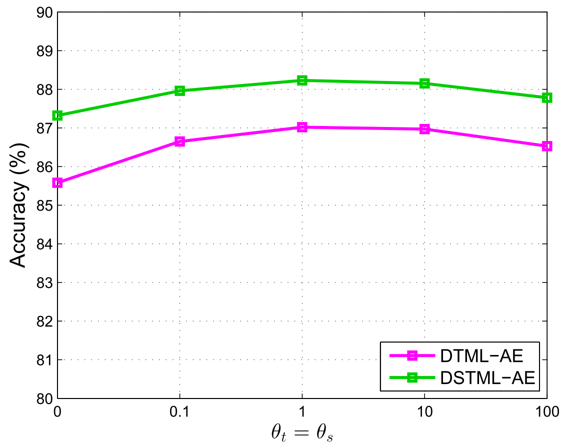


Fig. 6. The mean accuracy of the DTML-AE and DSTML-AE methods with varying values of θ_t and θ_s ($\theta_t = \theta_s$) on the LFW dataset.

other parameters are fixed. In the experiments, we directly set $\theta_t = \theta_s$ to reduce the number of parameters in our methods. Figure 6 shows the mean accuracy of the DTML-AE and DSTML-AE methods with varying values of θ_t and θ_s on the LFW dataset. From this figure, we can see that our methods can achieve stable performance when θ_t and θ_s are selected from [0.1, 10], and θ_t, $\theta_s = 1$ reports the best mean accuracy. Thus, we simply set θ_t, $\theta_s = 1$ for other datasets in our experiments. When θ_t and θ_s reach to zero, the performance reduces dramatically because the autoencoder regularization is not performed. When θ_s and $\theta_s \longrightarrow \infty$, the autoencoder regularization term of the DTML-AE and DSTML-AE methods is emphasized while labeled information and MMD regularization term are weakened such that their performance progressively decreases.

E. Discussion

1) Motivation of DSTML: Since we aim to learn a deep distance metric to measure the similarity of samples, we expect that the similarity obtained in hidden layers is also useful. We make some constrains on the hidden layers to avoid useless outputs, so that we are able to directly influence the hidden layer update to favor highly discriminative distance metrics. Another motivation is to reduce the effect of vanishing and exploding gradient problem in training because each hidden layer has a direct supervision from ground truth.

2) DTML Versus CNN-Based Method: AlexNet [15] shows that the domain disparity is well removed by using deep feature representation on objects. Generally, the performance of CNN-based method is better than that of our DTML when the CNN model is trained on the large-scale labeled dataset (e.g., ImageNet). While CNN-based methods can learn robust feature representations, these methods require a large number of labeled data for training because there are extensive number of parameters to estimate. For some practical applications such as cross-modal face matching, it is not easy to collect such large number of labeled data for deep feature learning. In contrast, the number of parameters in our DTML is relatively small so that it can also work well for discriminative metric learning with small size of training samples. In addition, CNN focuses on learning generic feature representation for cross-domain recognition while our method aims to seek domain transfer similarity measure for cross-domain visual recognition. In this work, we present a deep similarity learning method to better measure the similarity of samples. Hence, our method is complementary to existing deep learning methods.

3) Hand-Crafted Feature Versus Raw Data: There are three reasons for us to choose Hand-crafted features such as LBP and histograms features: 1) These hand-crafted features have shown reasonably good performance in previous face verification and person re-identification applications; 2) We observe that hand-crafted features with the learned deep metrics can obtain the favorable performance on several datasets; and 3) Existing deep learning methods usually require a large number of labeled training data to learn deep model such as CNN model. Our model can be considered as the fully connected part of the CNN model and our model doesn't utilize the convolutional and pooling layers. With the simplification, our method doesn't require many labeled training data because the number of parameters in our model is reduced.

4) Batch Optimization Versus Stochastic Algorithm: There are two reasons that our methods employ a batch optimization method rather than stochastic gradient descent (SGD) algorithm: 1) our methods need to compute the intra-class compactness and the interclass separability of source domain; and 2) a single sample is usually noisy, while batch tend to average a little of the noise out. For the large-scale training data, we may try to employ minibatch SGD to made small enough to be computationally tractable.

VI. Conclusion

In this paper, we have proposed a deep transfer metric learning (DTML) method for cross-dataset visual recognition. By learning a set of hierarchical nonlinear transformations which transfer discriminative knowledge from the labeled source domain to the unlabeled target domain, the proposed method can achieve better performance than existing linear metric learning methods. To better exploit the discriminative information from the source domain, we have further developed a deeply supervised transfer metric learning (DSTML) method by exploiting discriminative information from both the hidden layer and the top output layer to further improve the recognition performance. To preserve the local manifold of input data points in the metric space, we have also introduced two new methods, DTML with autoencoder regularization (DTML-AE) and DSTML with autoencoder regularization (DSTML-AE). Experimental results are presented to show the effectiveness of the proposed methods.

References

[1] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inf. Proc. Syst.*, 2002, pp. 505–512.

[2] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[3] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[4] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1875–1882.

[5] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 331–345, Feb. 2014.

[6] J. Lu, G. Wang, and P. Moulin, "Localized multifeature metric learning for image-set-based face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 529–540, Mar. 2016.

[7] J. Lu, G. Wang, W. Deng, and K. Jia, "Reconstruction-based metric learning for unconstrained face verification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 79–89, Jan. 2015.

[8] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2041–2056, Oct. 2015.

[9] J. Lu, V. E. Liong, and J. Zhou, "Cost-sensitive local binary feature learning for facial age estimation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5356–5368, Dec. 2015.

[10] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 518–531, Mar. 2016.

[11] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[12] J. Hu, J. Lu, and Y.-P. Tan, "Deep metric learning for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, doi: 10.1109/TCSVT.2015.2477936,2015.

[13] D.-Y. Yeung and H. Chang, "A kernel approach for semisupervised metric learning," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 141–149, Jan. 2007.

[14] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 325–333.

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

[16] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Proc. Syst.*, 2013, pp. 2553–2561.

[17] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3361–3368.

[18] G. B. Huang, H. Lee, and E. G. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2518–2525.

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[22] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.

[23] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Deep nonlinear metric learning with independent subspace analysis for face verification," in *Proc. Int. Conf. Multimedia*, 2012, pp. 749–752.

[24] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 34–39.

[25] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1137–1145.

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

[27] 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, pp. 1076–1089, May 2014.

[28] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2013.

[29] L. Zhang and D. Zhang, "Robust visual knowledge transfer via extreme learning machine-based domain adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4959–4973, Oct. 2016.

[30] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[31] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov. 2005.

[32] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer SVM for video concept detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1375–1381.

[33] 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, Mar. 2012.

[34] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. AAAI Conf. Artif. Intell.*, 2008, pp. 677–682.

[35] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[36] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[37] T. Liu, D. Tao, M. Song, and S. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, doi: 10.1109/TPAMI.2016.2544314.

[38] Z. Zha, T. Mei, M. Wang, Z. Wang, and X. Hua, "Robust distance metric learning with auxiliary knowledge," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 1327–1332.

[39] Y. Zhang and D.-Y. Yeung, "Transfer metric learning by learning task relationships," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1199–1208.

[40] Y. Zhang and D.-Y. Yeung, "Transfer metric learning with semi-supervised extension," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, May 2012, Art. no. 54.

[41] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Proc. ACCV Comput. Vis.*, 2012, pp. 31–44.

[42] Y. Luo, T. Liu, D. Tao, and C. Xu, "Decomposition-based transfer distance metric learning for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3789–3801, Sep. 2014.

[43] Z. Al-Halah, L. Rybok, and R. Stiefelhagen, "Transfer metric learning for action similarity using high-level semantics," *Pattern Recognit. Lett.*, vol. 72, pp. 82–90, Mar. 2016.

[44] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[45] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Research*, vol. 15, no. 1, pp. 3563–3593, Jan. 2014.

[46] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, 2007.

[47] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 566–579.

[48] W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by relative distance comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 653–668, Mar. 2013.

[49] A. J. Ma, P. C. Yuen, and J. Li, "Domain transfer support vector ranking for person re-identification without target camera label information," in *Proc. IEEE Int. Conf. Comput. Vis.*, Mar. 2013, pp. 3567–3574.

[50] F. Xiong, M. Gou, O. I. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 1–16.

[51] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 262–275.

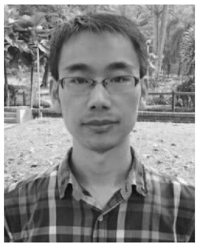
[52] W.-S. Zheng, S. Gong, and T. Xiang, "Associating groups of people," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2009, pp. 1–11.

[53] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.

[54] D. Baltieri, R. Vezzani, and R. Cucchiara, "3DPeS: 3D people dataset for surveillance and forensics," in *Proc. 1st Int. ACM Workshop Human Gesture Behavior Understand.*, 2011, pp. 59–64.

[55] L. Zhang, W. Zuo, and D. Zhang, "LSDT: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1177–1191, Mar. 2016.

[56] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.
[57] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2066–2073.



Junlin Hu received the B.Eng. degree from the Xi'an University of Technology, Xi'an, China, in 2008, and the M.Eng. degree from Beijing Normal University, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include computer vision, pattern recognition, and biometrics.

Jiwen Lu (M'11–SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. From 2011 to 2015, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision, pattern recognition, and machine learning. He has authored or co-authored over 130 scientific papers in these areas, where 32 were the IEEE Transactions papers. He is the Workshop Chair/Special Session Chair/Area Chair for over ten international conferences. He was a recipient of the National 1000 Young Talents Plan Program in 2015. He serves as an Associate Editor of the *Pattern Recognition Letters*, the *Neurocomputing*, and the IEEE ACCESS, a Managing Guest Editor of *Pattern Recognition* and *Image and Vision Computing*, a Guest Editor of *Computer Vision and Image Understanding* and *Neurocomputing*, and an Elected Member of the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society.

Yap-Peng Tan (M'97–SM'04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1993, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1995 and 1997, respectively, all in electrical engineering. From 1997 to 1999, he was with Intel Corporation, Chandler, AZ, and the Sharp Laboratories of America, Camas, WA. In 1999, he joined the Nanyang Technological University of Singapore, where he is currently an Associate Professor and the Associate Chair (Academic) with the School of Electrical and Electronic Engineering. His current research interests include image and video processing, content-based multimedia analysis, computer vision, and pattern recognition. He was the General Co-Chair of the 2015 IEEE Conference on Visual Communications and Image Processing and the 2010 IEEE International Conference on Multimedia and Expo. He served as the Chair of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society, a member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society, and a Voting Member of the ICME Steering Committee. He is an Editorial Board Member of the IEEE TRANSACTIONS ON MULTIMEDIA, the *EURASIP Journal on Advances in Signal Processing*, and the *EURASIP Journal on Image and Video Processing*, and an Associate Editor of the *Journal of Signal Processing Systems*.

Jie Zhou (M'01–SM'04) received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. In recent years, he has authored over 100 papers in peer-reviewed journals and conferences. Among them, over 40 papers have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and CVPR. His current research interests include computer vision, pattern recognition, and image processing. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor of the *International Journal of Robotics and Automation* and two other journals.