Deep Transfer Low-Rank Coding for Cross-Domain Learning

Zhengming Ding, *Member, IEEE*, and Yun Fu, *Senior Member, IEEE*

***Abstract*— Transfer learning has attracted great attention to facilitate the sparsely labeled or unlabeled target learning by leveraging previously well-established source domain through knowledge transfer. Recent activities on transfer learning attempt to build deep architectures to better fight off cross-domain divergences by extracting more effective features. However, its generalizability would decrease greatly due to the domain mismatch enlarges, particularly at the top layers. In this paper, we develop a novel deep transfer low-rank coding based on deep convolutional neural networks, where we investigate multilayer low-rank coding at the top task-specific layers. Specifically, multilayer common dictionaries shared across two domains are obtained to bridge the domain gap such that more enriched domain-invariant knowledge can be captured through a layerwise fashion. With rank minimization on the new codings, our model manages to preserve the global structures across source and target, and thus, similar samples of two domains tend to gather together for effective knowledge transfer. Furthermore, domain/classwise adaption terms are integrated to guide the effective coding optimization in a semisupervised manner, so the marginal and conditional disparities of two domains will be alleviated. Experimental results on three visual domain adaptation benchmarks verify the effectiveness of our proposed approach on boosting the recognition performance for the target domain, by comparing it with other state-of-the-art deep transfer learning.**

***Index Terms*— Deep learning, low-rank coding, transfer learning.**

I. Introduction

IN REALITY, there is always a situation that we can be accessible to the abundant unlabeled data while sparsely to labeled or no labeled data in the target domain. However, it is very time consuming and expensive to manually annotate the data. Transfer learning [1], [2] has achieved appealing performance in fighting off such a challenge through knowledge adaptation from an auxiliary well-labeled source domain, but with a different distribution [3]–[16]. Recently, hundreds of transfer learning techniques have been proposed, e.g., domain-invariant feature learning and classifier adaptation. Specifically, it is the most popular strategy to extract new valid features for the data from two domains with domain mismatch mitigated. Along this line, transfer sparse coding and low-rank coding manage to seek effective and domain-invariant representations for two domains to solve the domain shift [17]–[19]. Transfer subspace learning is also widely adopted to both handle the curse of dimensionality and domain mismatch [20]–[22]. Moreover, dictionary learning is one promising technique to extract effective representations from the original data, which has been widely adopted in transfer learning [3], [17], [23], [24]. However, the current dictionary-based transfer learning either builds a common dictionary in a latent space [3], [17], [25] or constructs a series of dictionaries [23] to bridge the source and target domains. They all deploy shallow structures, which are difficult to find common information across two domains, since they are unable to uncover the complex yet rich knowledge behind the data.

Recent research efforts have revealed that deep structure learning is able to generate more domain-invariant features for knowledge transfer with promising performance on existing cross-domain benchmarks [26]–[28]. Specifically, deep structure learning manages to unfold exploratory factors of variations within the data, and cluster representations layer by layer according to their similarity [29]. However, feature transferability will drop significantly in the top task-specific layers with domain discrepancy enlarged [7]–[9]. That is to say, the features extracted in the top task-specific layers will highly depend on the specific domains, which are not valid when facing novel domains.

Most recently, deep transfer learning aims to unify knowledge transfer and deep feature extraction into one training procedure [7]–[9], [30]–[32]. Thus, the classifier could be obtained with domain-invariant and informative representations. Thus, the learned feed-forward networks can be generalized to the unlabeled target by removing the distribution gap. However, these algorithms only consider the marginal distribution divergence across two domains, while ignoring the classwise intrinsic information of two domains. Furthermore, most existing transfer learning manages to assign each target sample with one hard pseudolabel to mitigate the classwise distribution mismatch [33], which may harm the knowledge adaptation if the target sample is annotated incorrectly. When data samples of two classes have distribution overlap, hard label assignment would destroy the intrinsic data structure.

Manuscript received March 29, 2018; revised June 7, 2018 and September 7, 2018; accepted October 3, 2018. Date of publication October 29, 2018; date of current version May 23, 2019. This work was supported in part by NSF IIS Award under Grant 1651902 and in part by the U.S. Army Research Office Award under Grant W911NF-17-1-0367. *(Corresponding author: Zhengming Ding.)*

Z. Ding is with the Department of Computer, Information and Technology, Indiana University—Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: zd2@iu.edu).

Y. Fu is with the Department of Electrical and Computer Engineering, College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

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/TNNLS.2018.2874567

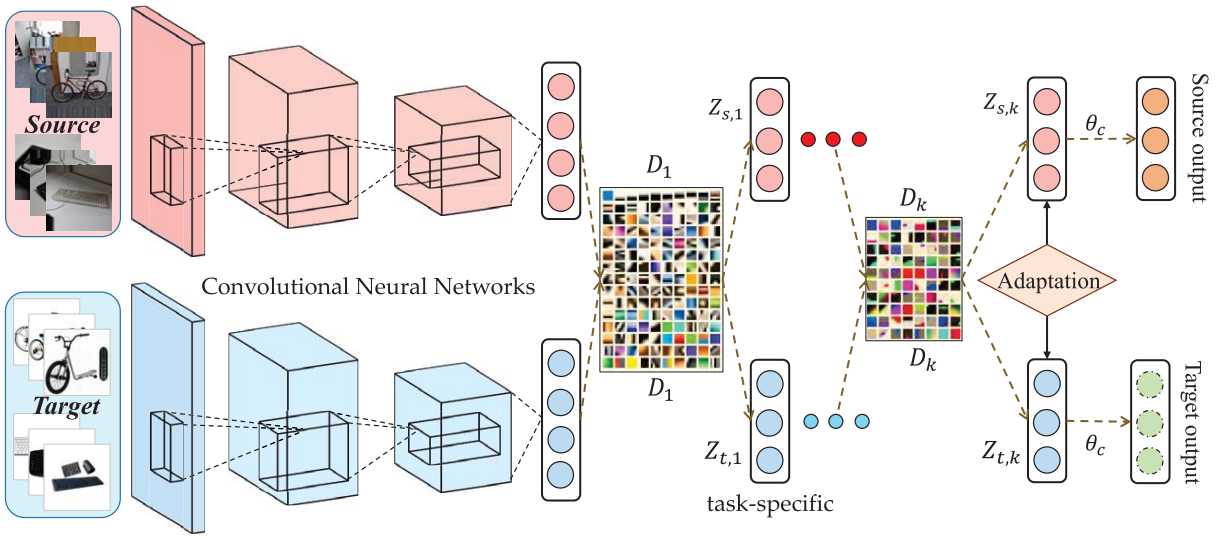


Fig. 1. Framework of our end-to-end DTLC: 1) convolutional layers are general and shared by two domains and 2) fully connected layers are *task specific*; hence, they are not transferable and should be adapted with multilayer low-rank coding. Specifically, multilayer dictionaries $D_i (i = 1, \ldots, k)$ are built to bridge the domain shift. The ith layer low-rank coding $Z_i = [Z_{s,i}, Z_{t,i}]$ would further generate a common dictionary D_{i+1} and new low-rank coding $Z_{i+1} = [Z_{s,i+1}, Z_{t,i+1}]$. With multilayer factorization, the distribution divergence across two domains tends to be reduced, and thus more well-labeled source knowledge could be transferred to the target. Furthermore, we develop a domain/classwise adaptation to guide the low-rank coding learning in a semisupervised manner.

In this paper, we present a novel deep transfer low-rank coding (DTLC) framework (Fig. 1) based on convolutional neural networks (CNNs), whose core idea is to build multilayer collective dictionaries to generate more discriminative domain-free low-rank coding to alleviate the target learning with the knowledge of the labeled source. Specifically, we extend the traditional low-rank coding with one common dictionary to multilayer dictionaries. To our best knowledge, we are the first to joint multiple latent dictionaries and low-rank coding to guide knowledge transfer for cross-domain learning. To sum up, we highlight our threefold contributions as follows.

1) Deep structures based on CNNs are constructed to extract more effective features from source and target through multilayer latent dictionaries along with the task-specific layers. In this way, we could learn more discriminative domain-invariant low-rank coding in a layerwise scheme to capture more complex information across two domains.
2) Rank minimization is exploited to guide the multilayer representations of two domains. With deeper structures, the low-rank constraint would cluster similar samples across two domains together. Therefore, the underlying intrinsic structures of two domains would be well matched for effective knowledge transfer.
3) Domain/classwise adaption scheme is investigated to further guide the deep low-rank coding learning in a semisupervised fashion, which substantially enhances the adaptation ability. To be specific, we assign each target sample to multiple classes with different probabilities, and then we design a novel loss function to effectively transfer well-labeled source knowledge.

II. Related Work

In this section, we briefly review three lines of works, i.e., sparse/low-rank coding, transfer learning, and further deep transfer learning.

A. Sparse/Low-Rank Coding

Given a data matrix $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}$, with m data samples lying in the feature space with dimensionality d. Consider $D = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n] \in \mathbb{R}^{d \times n}$ as the dictionary in which $\boldsymbol{d}_i$ denotes a basis vector. $Z = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m] \in \mathbb{R}^{n \times m}$ means the representation matrix, in which each $\boldsymbol{z}_i$ is the new feature representation for $\boldsymbol{x}_i$.

The aim of sparse/low-rank coding is to seek a dictionary and its corresponding coefficients, and thus, the original data could be efficiently approximated [34]. Generally, the reconstruction error is used to approximate any given data samples X with the dictionary D and new codes Z, shown as follows:

$$\begin{aligned} &\min_{D,Z} \ \mathcal{N}(Z) + \lambda \|X - DZ\|_{\mathrm{F}}^2 \\ &\text{s.t.} \ \|\boldsymbol{d}_i\|_2^2 \leq 1 \quad \forall i = 1, \ldots, n \end{aligned} \tag{1}$$

where $\mathcal{N}(\cdot)$ denotes different kinds of constraints, e.g., Frobenius norm ($\|\cdot\|_{\mathrm{F}}$), sparse constraints (e.g., $l_1, l_{2,1}$-norm [17]), and nuclear norm ($\|\cdot\|_*$) [25], [35]. $\|\boldsymbol{d}_i\|_2^2 \leq 1$ is used to control the complexity of the dictionary model. D and Z could be optimized alternatively by fixing one variable and optimizing the other variable until it converges.

B. Transfer Learning

Transfer learning has been treated as an appealing approach for lots of real-world problems with a limited labeled

data challenge. In general, transfer learning is proposed to handle this challenge by borrowing knowledge from other external well-labeled data [1], [2]. The key in transfer learning is to deal with the domain mismatch to benefit from the labeled source knowledge [3], [18], [36]. So far, there are plentiful transfer learning models proposed, including domain-invariant feature learning [3], [23] and classifier parameter adaptation [19], [37]–[40].

Specifically, domain-invariant feature learning is one of the most popular strategies, which attempts to learn a common feature space in which the domain shift is mitigated. Following this, there are several techniques well explored to align two different domains, e.g., subspace learning, nonlinear projections, and dictionary learning. Among them, dictionary learning scheme manages to construct a single shared dictionary or multiple common dictionaries to further seek domain-invariant new representation [3], [23]. Various kinds of regularizers are investigated on the new representation to preserve specific properties, e.g., sparse constraint, rank constraint, or locality constraint. Along this line, the previous cross-domain learning explored the low-rank constraint to capture the shared information across different tasks. Yang *et al.* [41] proposed multiple feature selection functions across different tasks and adopted a low-rank constraint to seek the common structure from different related tasks. In this paper, we explore dictionary learning through the low-rank constraint. Different from previous work, we propose to seek multilayer latent shared dictionaries by further factorizing low-rank codings so that more shared knowledge could be uncovered. Through deep structures based on CNNs, more complex abstraction of two domains could be captured to handle the marginal and conditional distributions of two domains.

C. Deep Transfer Learning

Most recently, deep transfer learning has been widely explored to unify the deep structure learning and knowledge transfer into one framework [7]–[9], [31]. The idea behind is to incorporate domain alignment strategies at the top layers to explicitly solve the enlarged domain discrepancy resulted from traditional deep learning models. Generally, maximum mean discrepancy (MMD) loss or revised MMD loss [7], [42] and adversarial loss [8] are two kinds of popular strategies to align different domains during deep structure learning. However, current deep domain adaptation algorithms mainly explore to couple two domains, each as a whole, and thus ignore the conditional distribution divergence across two domains. Therefore, it is more appropriate to incorporate classwise alignment for effective feature learning.

Differently, our work investigates multilayer dictionaries both in linear version and nonlinear version to build deep structures for effective domain-invariant feature learning. This paper is the extension of our previous conference [18]. The major difference is that we jointly learn multilayer dictionaries to achieve the final low-rank coding. Our previous work [18] exploits a stacked strategy to construct a hierarchical structure, where the output low-rank coding of the previous layer would be as the input of the next layer. In this way, the low-rank coding is built in a stacked-layer fashion so that they are not jointly optimized. In our extended version, we adapt our multilayer dictionary decomposition to CNN architecture. Besides, our classwise adaptation plays the same role as the iterative structure term in our previous work [18] as both aim to mitigate the conditional distribution differences across two domains in a semisupervised fashion. Specifically, the soft labels of the target would be optimized iteratively so that it could boost the discriminative low-rank coding learning. In this extension, we explore the nonlinear version of deep low-rank coding in order to capture more shared knowledge across two domains.

III. Deep Transfer Low-Rank Coding

In this section, we first provide preliminary knowledge and motivation, then present the developed deep low-rank coding for knowledge transfer through multilayer latent dictionaries in two versions. Finally, we provide analysis and discussion for better understanding the proposed model. Note that we use upper case character to denote the matrix, e.g., A, lower case character to represent the scale, e.g., a, and bold lower case character means the vector, e.g., $\boldsymbol{a}$.

A. Preliminaries and Motivation

For transfer learning, given the unlabeled target domain $\mathcal{T}$ with a set of m_t data points $X_t = \{\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,m_t}\}$ and the labeled source domain $\mathcal{S}$ with a set of m_s data points $\{X_s, Y_s\} = \{(\boldsymbol{x}_{s,1}, \boldsymbol{y}_{s,1}), \ldots, (\boldsymbol{x}_{s,m_s}, \boldsymbol{y}_{s,m_s})\}$. In this paper, we consider the source and target data are from C classes. X is the concatenate of source and target, i.e., $X = [X_s, X_t] \in \mathbb{R}^{d\times m}$, in which $m = m_s + m_t$ is the total size of two domains. The new representation $Z = [Z_s, Z_t]$ is also for source and target [17]. Suppose source and target tend to be characterized by probability distributions p and q, respectively, and we manage to seek domain-invariant features to mitigate the cross-domain gap, then train a classifier $\boldsymbol{y} = \Theta(\boldsymbol{x})$, which is able to reduce the target risk $\epsilon_t(\Theta) = \Pr_{(\boldsymbol{x},y)\sim q}[\Theta(\boldsymbol{x}) \neq \boldsymbol{y}]$.

In traditional transfer sparse/low-rank coding, such one single common dictionary shared by source and target is built to achieve new representations for two domains through mean embedding matching [17], [25]. However, such a single-layer sparse/low-rank coding cannot always exploit enough common knowledge across two domains to alleviate the target learning. In many scenarios, the data we desire to analyze are often complex and involve various factors, e.g., pose variance, illuminations, and low resolution. Recent research efforts on deep structure learning have witnessed to show promising performance in cross-domain learning [8], [18], [31], [43]–[45]. Inspired by this, we desire to embed knowledge transfer through multilayer common dictionaries during deep feature learning. Hence, a series of common dictionaries would be built to further decompose the previous learned low-rank coding and generate new low-rank coding. To this end, multilayer common dictionaries are able to consolidate the transferability from the complicated data structure of source and target domains.

Moreover, MMD [46] is a very popular strategy to measure the domain difference of both marginal and conditional distributions [21], [33], [47] by involving the pseudolabels of

the target data. Current domain adaptation approaches attempt to predict pseudolabels of the target data iteratively through some specific classifiers, and then feedback to refine the classwise MMD. Unfortunately, it would bring some problems if we only assign every target sample to one specific class label (named as "hard label"). For example, if a target sample was predicted with a wrong label, and the classifier would be trained problematically. On the other hand, the hard label could destroy the data structure when some classes are overlapped in the feature space. Another phenomenon in deep model training is that the recognition performance would be increased with more iterations' optimization. Thus, the probability of the true class label for unlabeled target data will increase to a higher value.

Considering these reasons, we assume every target sample to be annotated with all the categories in different probabilities, named as "soft label" [8]. That is to say, an incorrectly labeled data instance could still contribute to the classifier learning, since its probability for the true label does not equal to zero in most cases. To further build effective deep features, we proposed a novel classwise adaptation formula to supervise the knowledge transfer by utilizing the target soft labels. On the other hand, the classifier would be more discriminative with more domain-invariant features across two domains. These two strategies would trigger each other and benefit both.

B. Deep Transfer Low-Rank Coding

As we mentioned before, the dictionary learning technique manages to seek one common dictionary or a series of dictionaries to mitigate the distribution differences across the source and target domains [3], [17], [23], [25]. With different strategies, e.g., maximum marginal distribution, the newly learned representations Z_s and Z_t should have similar distributions. Specifically, $X_s \approx DZ_s$ and $X_t \approx DZ_t$, which is $X \approx DZ$, where $Z = [Z_s, Z_t]$. Generally, X_s and X_t have different distributions; therefore, such new codings Z_s, Z_t through a common dictionary D may still have distribution mismatch, especially when the original source and target are in largely different distributions.

Fortunately, the new learned codings Z_s and Z_t should have more similar distribution than X_s and X_t [17], [25]. Therefore, we aim to exploit multilayer dictionaries and codings learning to uncover more shared information across two domains. The multilayer matrix factorization is expressed as

$$\begin{aligned} X &\approx D_1 Z_1 \\ X &\approx D_1 D_2 Z_2 \\ &\vdots \\ X &\approx D_1 D_2 \dots D_k Z_k \end{aligned} \tag{2}$$

where $D_i (1 \le i < k) \in \mathbb{R}^{d_{i-1} \times d_i} (d_0 = d)$ and each new representation $Z_i (1 \le i < k) \in \mathbb{R}^{d_i \times m}$ across two domains is employed to build a new common dictionary then more similar representation of two domains would be generated. Note that we have the decomposition constraint as $Z_{k-1} = D_k Z_k$.

To this end, we propose our DTLC framework with a scale constraint on the dictionary atoms as follows:

$$\begin{aligned} \min_{D_1,\dots,D_k,Z_k} \ & \operatorname{rank}(Z_k) + \lambda \|X - D_1 D_2 \dots D_k Z_k\|_{\mathrm{F}}^2 \\ \text{s.t.} \ & \left\|\boldsymbol{d}_i^j\right\|_2^2 \le 1 \quad \forall i, j \end{aligned} \tag{3}$$

where $\boldsymbol{d}_i^j$ is the jth atom of D_i, $\operatorname{rank}(\cdot)$ is the rank operator of a matrix, and λ is the tradeoff parameter between two terms.

Remark: With a low-rank constraint on the last layer coding Z_k, we could achieve that the latent intermediate codings Z_i are all low rank. Since we have $Z_{k-1} = D_k Z_k$, we could observe that $\operatorname{rank}(Z_{k-1}) = \operatorname{rank}(D_k Z_k)$. As we know that $\operatorname{rank}(D_k Z_k) \le \min\big(\operatorname{rank}(Z_k), \operatorname{rank}(D_k)\big)$, and thus $\operatorname{rank}(Z_{k-1}) \le \operatorname{rank}(Z_k)$. Hence, we could deduct that $\operatorname{rank}(Z_1) \le \operatorname{rank}(Z_2) \le \cdots \le \operatorname{rank}(Z_k)$. That is, our low-rank constraint on the last layer Z_k would preserve the low-rank property on the coding of each layer. In this way, low-rank coding in each layer would uncover the global structure of two domains.

Our hypothesis is that by further learning common latent dictionaries and low-rank codings, we can build a deep structure model, which could first automatically uncover the latent hierarchical shared knowledge across two domains; second, more effective representations of the data would be learned to better mitigate the distribution divergence of two domains in a layerwise fashion so that the final layer coding would have lower variability to two domains. Using low-rank constraint for cross-domain learning has been explored in previous works, e.g., feature selection in the multimedia analysis by capturing common information across different tasks. The advantage of our model is we explore a multilayer scheme to uncover more classwise structure across two domains.

1) Semisupervised Knowledge Adaptation: To make low-rank coding robust to various probability distributions, we would expect that the dictionary is able to capture the commonality within the labeled source and unlabeled target. However, although we have extracted the k-layer low-rank coding, the disparity of two domains may still not be well tackled without any supervised guidance during model training.

In transfer learning, it is very important to mitigate the distribution difference and a natural way is to pull the probability distributions of two domains close to each other in the learned new low-rank space. In other words, for the extracted low-rank features $Z_k = [Z_k^s, Z_k^t]$ (Z_k^s for source and Z_k^t for the target) from original X, the probability distributions of source and target samples should be as close as possible.

a) Domainwise Adaptation: Since it is nontrivial to directly estimate probability densities, we explore the popular strategy, empirical MMD [7] to measure the distance. Specifically, MMD compares different distributions through the distance between the sample centers of two domains in the d_k-dimensional low-rank feature space, namely,

$$\begin{aligned} \mathcal{M}(Z_k) &= \left\| \frac{1}{m_s} \sum_{i=1}^{m_s} \boldsymbol{z}_{k,i} - \frac{1}{m_t} \sum_{j=m_s+1}^{m} \boldsymbol{z}_{k,j} \right\|_2^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} \boldsymbol{z}_{k,i}^{\top} \boldsymbol{z}_{k,j} W_{ij} = \operatorname{tr}\big(Z_k W Z_k^{\top}\big) \end{aligned} \tag{4}$$

in which $z_{k,i/j}$ is the i/jth column of Z_k and $\mathrm{tr}(\cdot)$ is the trace operator of a matrix while W is the MMD matrix, whose element is calculated as follows:

$$W_{ij} = \begin{cases} \frac{1}{m_s^2}, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S} \\ \frac{1}{m_t^2}, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{T} \\ \frac{-1}{m_s m_t}, & \text{otherwise.} \end{cases}$$

b) Semisupervised Classwise Adaptation: Previous MMD (4) only reduces the disparity in the marginal distributions cannot guarantee to well fight off the conditional distribution divergence of two domains. Actually, it is essential to mitigate the conditional distribution difference across different domains for effective knowledge transfer. Unfortunately, the target data are totally unlabeled or limited labeled, so that it is nontrivial to align the conditional distributions. However, we adopt the soft labels of target data by annotating each target instance to multiple classes with various weights.

Suppose $\boldsymbol{p}_j \in \mathbb{R}^C$ is the soft label for the jth target instance, where each element $p_{c,j}$ denotes the probability of the jth unlabeled target sample to be annotated to the cth class ($p_{c,j} \geq 0$ and $\sum_{c=1}^{C} p_{c,j} = 1$). In other word, we can consider that each target partially contributes to different classes during knowledge transfer. To this end, we propose a semisupervised classwise adaptation term with soft labels to align source and target domains, which minimizes the classwise centers across the source and target domains. Since target data are unlabeled, and thus we do not hope to constrain one target sample to be close to one source class center with the same label [33], [48]. Instead, we aim to enforce one target instance to be close to multiple source class centers with various probabilities (obtained from the Softmax output), which would preserve a good target structure during adaptation and avoid an invalid alignment across the source and target domains. From another view, we aim to use the cross-domain center reconstruction loss to guide the unlabeled target samples' label output, similar to Softmax loss for labeled source data.

To enhance the adaptation ability iteratively, we should update the probabilistic label of each target instance. Hopefully, the probability for the true label would be close to 1. To this end, we can deal with the conditional distribution divergences across source and target by designing a novel classwise adaption loss as follows:

$$\mathcal{C}(Z_k) = \sum_{c=1}^{C} \left\| \frac{1}{m_s^c} \sum_{i=1}^{m_s^c} \boldsymbol{z}_{k,i}^s - \frac{1}{m_t^c} \sum_{j=1}^{m_t} p_{c,j} \boldsymbol{z}_{k,j}^t \right\|_2^2 \tag{5}$$

where m_s^c means the source class size of the cth class. m_t^c represents the target class size of the cth class, which is not an integer and not directly given, since we cannot get the true target class size per class. Hence, we can calculate m_t^c in this way, i.e., $m_t^c = \sum_{j=1}^{m_t} p_{c,j}$. Hence, we can deduct that

$$\mathcal{C}(Z_k) = \sum_{c=1}^{C} \mathrm{tr}\left(Z_k W^{(c)} Z_k^\top\right) \tag{6}$$

where $W^{(c)}$ is the revised MMD matrix for class c, which is defined in the following way:

$$W_{ij}^{(c)} = \begin{cases} \frac{1}{m_s^{(c)} m_s^{(c)}}, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}^{(c)} \\ \frac{p_{c,j} p_{c,i}}{m_t^{(c)} m_t^{(c)}}, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{T} \\ \frac{-p_{c,j/i}}{m_s^{(c)} m_t^{(c)}}, & \text{if } \boldsymbol{x}_{i/j} \in \mathcal{S}^{(c)}, \ \boldsymbol{x}_{j/i} \in \mathcal{T} \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{S}^{(c)}$ is the source domain for class c.

To build an end-to-end deep architecture, we adopt the Softmax classifier loss as the final layer

$$\mathcal{J}(Z_k, \Theta, Y) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c=1}^{C} y_{c,i} \log \frac{e^{\boldsymbol{\theta}_c^\top z_{k,i}}}{\sum_{u=1}^{C} e^{\boldsymbol{\theta}_u^\top z_{k,i}}}$$

where $y_{c,i}$ is the label probability the sample $z_{k,i}$ was assigned to class c. Specifically, $y_{c,i}$ for source part is fixed and for target part is iteratively updated during model training.

Thus, we formalize our final objective function $\mathcal{L}$ with the constraint ($\|\boldsymbol{d}_i^j\|_2^2 \leq 1, \forall i, j$) as

$$\begin{aligned} \mathcal{L} \ = \ & \mathcal{J}(Z_k, \Theta, Y) + \lambda \|X - D_1 D_2 \dots D_k Z_k\|_{\mathrm{F}}^2 \\ & + \alpha \sum_{c=0}^{C} \mathrm{tr}\left(Z_k W^{(c)} Z_k^\top\right) + \beta \|Z_k - AB\|_{\mathrm{F}}^2 \end{aligned} \tag{7}$$

where we further achieve $W^{(0)} = W$ when substituting $m_s^{(0)} = m_s, m_t^{(0)} = m_t, \mathcal{S}^{(0)} = \mathcal{S}, \mathcal{T}^{(0)} = \mathcal{T}$. α and β are the balanced parameters. Moreover, rank minimization is an nondeterministic polynomial-time-hard problem and nuclear norm is a good alter-native [18], [35], which, however, is very time consuming. Motivated by recent work [49], the incorporation of explicit rank control would result in a more efficient optimization algorithm. Actually, we hope samples from the same class ideally tend to be spanned by only one basis. Since we have access to the true rank of Z_k, i.e., the class size C, and thus, we have $Z_k \approx AB$ to replace the rank minimization (the forth term), where $A \in \mathbb{R}^{d_k \times C}$ and $B \in \mathbb{R}^{C \times n}$.

As we mentioned before, soft labels are optimized iteratively when we achieve low-rank coding Z_k and a classifier parameter. Here, we have $p_{c,j} = ((e^{\boldsymbol{\theta}_c^\top z_{k,j}^t})/(\sum_{u=1}^{C} e^{\boldsymbol{\theta}_u^\top z_{k,j}^t}))$, where $\boldsymbol{\theta}_c$ is the cth column of Θ and $z_{k,j}^t$ is the jth column of coding Z_k^t for the target domain.

2) Nonlinear Extension: By learning multilayer linear dictionaries from the original data with complex distributions, we may fail to illustrate the nonlinear structures efficiently, which happen in the latent common knowledge shared by two domains. Inspired by the nonlinear activations in neural networks, we could also introduce nonlinear activations between any two layers. This practice enables us to extract more efficient low-rank codings that are nonlinearly separable in the original input space.

From neurophysiology paradigms, the theoretical and experimental evidence indicates that the human visual system has a hierarchical and rather nonlinear scheme [50] in the processing data structure, where neurons become selective to process progressively more complex features of the data structure.

Furthermore, Malo *et al.* [51] argued that employing an adaptive nonlinear data representation algorithm would lead to a reduction of the statistical and the perceptual redundancy amongst the representation elements.

From the view of mathematical point, we can adopt nonlinear functions $f(\cdot)$ to guide every implicit low-rank coding $Z_1, \ldots, Z_{k-1}$. In this way, we could better approximate the nonlinear intrinsic manifold structure of the original data matrix X. In other words, we could improve the expressibility of our framework and allow for a better reconstruction of the original data through nonlinear activations. This has been verified by the Stone–Weierstrass theorem [52].

To incorporate nonlinearities to our framework, we modify the ith low-rank codings Z_i, by setting

$$Z_i \approx f(D_{i+1}Z_{i+1}) \tag{8}$$

which in turn converts the final objective function (3) of the model $\mathcal{L}$ with the constraint $(\|\boldsymbol{d}_i^j\|_2^2 \leq 1, \forall i, j)$ as

$$\begin{aligned}\mathcal{L} &= \mathcal{J}(Z_k, \Theta, Y) + \lambda\|X - D_1 f(D_2 f(\cdots f(D_k Z_k)))\|_{\mathrm{F}}^2 \\ &\quad + \alpha \sum_{c=0}^{C} \mathrm{tr}\left(Z_k W^{(c)} Z_k^\top\right) + \beta\|Z_k - AB\|_{\mathrm{F}}^2 \end{aligned} \tag{9}$$

where we adopt the *ReLU* activation function for nonlinear version, following most deep learning works for better convergence [7], [9], [45]. Note that the low-rank property cannot preserve when we adopt nonlinear activation functions, e.g., *Sigmoid*, *TanHyperbolic*, and *ReLU*. It would be an open problem to discuss the rank property for nonlinear activation functions. In this paper, we mainly focus on the low-rank property transit for linear version, and nonlinear version is just the generalization to linear version. Furthermore, our goal is to explore linear/nonlinear mappings to guide the feature learning and we focus on achieving low-rank coding in the final-layer representation, and therefore, we did not pay more attention to the low-rank property of each layer.

3) Model Training: Targeting at expediting the convergence of the multiple dictionaries and low-rank coding for our model, we first pretrain each layer to obtain initial approximations for D_i, Z_i. This strategy would greatly cut down the computational cost in the training stage. This is a tactic, which has successfully used in deep autoencoder networks [53]. Specifically, for a pretraining step, we first learn the dictionary D_1 and low-rank coding Z_1 from the initial data matrix X via (1). Following this, we further learn dictionary D_2 and low-rank coding Z_2 from the first-layer low-rank codings Z_1, keeping doing so until we have pretrained all dictionaries and low-rank coding. Furthermore, to initialize each dictionary, we exploit the K-SVD method to achieve an effective initialization [54], that is, the initial subdictionary D_i^j for class j is obtained. The input dictionary D_i is pretrained on the source domain Z_{i-1} by integrating all subdictionaries for each class, i.e., $D_i = [D_i^1, D_i^2, \ldots, D_i^C]$ (C is the class number).

Moreover, as we mentioned before, the target data are unlabeled and assigned with soft labels $p_{c,j}$ for classwise adaptation. We use the pretrain multilayer structure to achieve the soft labels of the target data. Furthermore, we propose to address the problem in two subproblems. First of all, we fix $\boldsymbol{p}_j$, m_t^c, and $W^{(c)}$, then optimize multiple dictionaries D_i, the low-rank coding Z_k, and the classifier parameters Θ. Second, we update the soft label of target data $\boldsymbol{p}_j$ as well as the new target class size m_t^c and $W^{(c)}$, then feedback to optimize the model parameters. To achieve better soft labels, we update the second step after a specific iteration of the first step. In this way, we could iteratively solve two subproblems until the optimization converges. In general, better feature representations would generate more accurate $\boldsymbol{p}_j$, m_t^c, and $W^{(c)}$. On the other hand, more accurate $\boldsymbol{p}_j$, m_t^c, and $W^{(c)}$ would trigger the training of networks. These two subproblems converge well and we show the convergence curves in the experiments.

For the first subproblem, we propose an alternating optimization approach to iteratively learn the low-rank codings Z_k, A, and B, and dictionaries D_i and Θ. Here, we take the nonlinear version as an example, since the linear version is its special case when $f(\cdot)$ is an identity function. It is easy to notice that (7) is smooth and twice-differentiable. Hence, we can still explore a stochastic gradient descent to solve this unconstrained optimization problem.

a) Multilayer Dictionary Learning: In order to obtain the derivative for the ith dictionary $(1 < i \leq k)$, we exploit the chain rule and have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial D_i} &= \frac{\partial \mathcal{L}}{\partial (D_i Z_i)} Z_i^\top = \left[\frac{\partial \mathcal{L}}{\partial f(D_i Z_i)} \odot \nabla f(D_i Z_i)\right] Z_i^\top \\ &= \left[\frac{\partial \mathcal{L}}{\partial Z_{i-1}} \odot \nabla f(D_i Z_i)\right] Z_i^\top \end{aligned} \tag{10}$$

where $\odot$ is the elementwise multiplication and $\nabla f(D_i Z_i)$ is the first derivative of function $f(\cdot)$ with respect to $D_i Z_i$. Specifically, the derivation of the first dictionary D_1 is then identical to the version of the framework with one layer as

$$\frac{\partial \mathcal{L}}{\partial D_1} = \frac{\lambda}{2}(X - D_1 Z_1) Z_1^\top. \tag{11}$$

For the optimization of the dictionaries, we add a normalization per atom in the dictionary to constrain its scale [17].

b) Low-Rank Coding Learning: While the derivative for the ith low-rank codings $(1 < i < k)$, we get

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial Z_i} &= D_i^\top \frac{\partial \mathcal{L}}{\partial (D_i Z_i)} = D_i^\top \left[\frac{\partial \mathcal{L}}{\partial f(D_i Z_i)} \odot \nabla f(D_i Z_i)\right] \\ &= D_i^\top \left[\frac{\partial \mathcal{L}}{\partial Z_{i-1}} \odot \nabla f(D_i Z_i)\right] \end{aligned} \tag{12}$$

and specifically for the first and last layers, we have

$$\frac{\partial \mathcal{L}}{\partial Z_1} = \frac{\lambda}{2} D_1^\top (X - D_1 Z_1) \tag{13}$$

and

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial Z_k} &= D_k^\top \left[\frac{\partial \mathcal{L}}{\partial Z_{k-1}} \odot \nabla f(D_k Z_k)\right] + 2\alpha \sum_{c=0}^{C} Z_k W^{(c)} \\ &\quad + \frac{\partial \mathcal{J}(Z_k, \Theta, Y)}{\partial Z_k} + \beta(Z_k - AB) \end{aligned} \tag{14}$$

where $((\partial \mathcal{J}(Z_k, \Theta, Y))/\partial Z_k)$ is a $d_k \times m$ matrix with the ith column vector as $-(1/m)(\sum_{c=1}^{C}(\theta_c - ((\sum_{u=1}^{C} \boldsymbol{\theta}_u e^{\boldsymbol{\theta}_u^\top z_{k,i}})/$

$(\sum_{u=1}^{C} e^{\boldsymbol{\theta}_u^\top z_{k,i}}))))$. For A

$$\frac{\partial \mathcal{L}}{\partial A} = Z_k B^\top - ABB^\top = 0, \quad \Rightarrow A = Z_k B^\top (BB^\top)^\dagger \tag{15}$$

while for B

$$\frac{\partial \mathcal{L}}{\partial B} = A^\top AB - A^\top Z_k = 0, \quad \Rightarrow B = (AA^\top)^\dagger A^\top Z_k \tag{16}$$

where † is the Moore–Penrose pseudoinverse.

c) Classifier Parameter Training: Since we adopt a standard Softmax loss with the input of low-rank coding Z_k, we could calculate the gradient descent of Θ as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_c} = \frac{\partial \mathcal{J}(Z_k, \Theta, Y)}{\partial \boldsymbol{\theta}_c} = -\frac{1}{m} \sum_{i=1}^{m} \Phi_i z_{k,i}^\top \tag{17}$$

where $\Phi_i = \sum_{c=1}^{C} y_{c,i}(1 - ((e^{\boldsymbol{\theta}_c^\top z_{k,i}})/(\sum_{u=1}^{C} e^{\boldsymbol{\theta}_u^\top z_{k,i}})))$.

C. Deep Transferable Networks

In this section, we propose to jointly learn our multilayer low-rank coding and CNNs parameters in an end-to-end manner.

In our DTLC architecture, the partial derivatives with respect to D_i, Z_k, A, B, and Θ are defined in Sections III-A and III-B. We could further calculate the partial derivatives with respect to X as

$$\frac{\partial \mathcal{L}}{\partial X} = 2\lambda(X - \mathcal{R}(D_1, D_2, \ldots, Z_k)) \tag{18}$$

where $\mathcal{R}(D_1, D_2, \ldots, Z_k) = D_1 D_2 \ldots D_k Z_k$ for the linear version and $\mathcal{R}(D_1, D_2, \ldots, Z_k) = D_1 f(D_2 f(\ldots f(D_k Z_k)))$ for the nonlinear version. Once we obtain $(\partial \mathcal{L}/\partial X)$, we then are able to perform the standard back propagation [55] to update the CNN parameters with input source and target images as well as source one-hot labels.

To train a powerful deep CNN model, the key challenge is that the target domain has no or just sparsely labeled samples, and hence, it is impossible to directly adapt CNN to the target domain via fine-tuning. In this way, we follow the previous deep transfer learning [7]–[9] to initialize the CNN parameters with some pretrained networks, e.g., AlexNet [26].

D. Model Comparison

There are related state-of-the-art deep transfer works to our proposed model, i.e., simultaneous deep transfer (SDT) [8], deep adaptation network (DAN) [7], reverse gradient (RevGrad) [9], D-COREL [31], and deep hashing network (DHN) [56]. Interestingly, SDT also proposed soft label of the target data to optimize the model training, which is similar to our idea by annotating each target instance to multiple classes with different probabilities. However, SDT neglected the conditional distribution differences between two domains, since it only used single-layer domain confusion to mitigate the marginal disparity of two domains. Although DAN adopted multilayer adaption through the multikernel strategy, it also only considered the domainwise mean embedding matching while ignoring the conditional distribution difference. DHN adopts the same structure and adaptation strategy to DAN. The key difference is DHN explores the hash coding in the final layer. RevGrad developed a novel gradient reversal layer, which ensured that the feature distributions across two domains are constrained to be similar, resulting in the domain-invariant features. However, RevGrad did not involve the conditional distribution into the gradient reversal layer. D-COREL adapts the COREL loss [10] to guide the deep knowledge transfer. To explore more discriminative knowledge across source and target domains, we tend to assign each target sample to multiple classes with different probabilities. In this way, we could build a novel classwise mean embedding matching as well as multilayer common dictionaries to transfer more well-labeled source knowledge during deep structure learning.

IV. Experimental Results

In this section, we evaluate on three cross-domain benchmarks to testify our designed model. First, we list the benchmarks description and experimental setup. Then, we compare our proposed model with several state-of-the-art domain adaptation and deep learning algorithms.

A. Data Sets and Experimental Setting

1) Data Sets Description: Office-31[1] [57] is a well-known cross-domain benchmark, which includes 4652 samples of 31 different objects from three various domains, i.e., Amazon (A), Webcam (W), and digital single lens reflex (DSLR) (D). To be specific, samples of Amazon are collected from www.amazon.com, while image samples of Webcam and DSLR are taken with a web camera and a digital single lens reflex camera under office environments, respectively. The major difference of Webcam and DSLR is the resolution gap. We perform six cross-domain tasks by randomly selecting two as a combination.

Office-10 + Caltech-10 [58] contains the 10 common objects shared across the Office-31 and Caltech-256 (C)[2] data sets. This data set has been widely used in the current domain adaptation methods [7], [18]. We can build 12 transfer tasks to evaluate our proposed algorithm. With more cross-domain learning tasks, we target an unbiased look over the domain bias.

Office + Home[3] [56] is a most recent cross-domain benchmark with more samples and objects. In total, there are 65 objects shared by four domains with about 15 500 image samples. Specifically, four domains are Art (artistic drawing objects), Clipart (images collected from www.clipart.com), Product (samples similar to Amazon almost with clean background), and Real-World (object images taken with regular cameras).

2) Comparisons: We experiment with the following state-of-the-art domain adaptation methods.

1) CORAL [10] is a "frustratingly easy" unsupervised domain adaptation approach that couples the second-order statistics across two domains' distributions with a linear projection.

[1] https://people.eecs.berkeley.edu/~jhoffman/domainadapt/#data_sets_code
[2] http://authors.library.caltech.edu/7694/
[3] https://hemanthdv.github.io/officehome-dataset/

TABLE I

AVERAGE RECOGNITION RATE (%) FOR UNSUPERVISED ADAPTATION ON OFFICE-31 DATA SET WITH FULL-SAMPLING PROTOCOL, WHERE A = AMAZON, D = DSLR, AND W = WEBCAM

Config	COREL [10]	CNN [26]	DDC [45]	RevGrad [9]	DAN [7]	D-CORAL [31]	DHN [56]	DLRC [18]	DTLC$_L$	DTLC$_N$
W→A	48.2±0.0	49.8±0.4	52.2±0.4	52.7±0.2	53.1±0.3	51.5±0.3	52.8±0.2	48.8±0.4	51.3±0.4	**53.9**±0.5
W→D	99.8±0.0	99.0±0.2	98.5±0.4	99.2±0.3	99.0±0.2	99.2±0.1	98.8±0.2	94.9±0.5	98.9±0.5	**99.3**±0.4
A→W	64.3±0.0	61.6±0.5	61.8±0.4	**73.0**±0.6	68.5±0.4	66.4±0.4	68.3±0.4	61.3±0.4	66.5±0.4	70.4±0.3
A→D	65.7±0.0	63.8±0.5	64.4±0.3	67.1±0.3	67.0±0.4	66.8±0.6	66.4±0.2	60.3±0.5	63.7±0.4	**68.2**±0.5
D→A	48.5±0.0	51.1±0.6	52.1±0.8	54.5±0.4	54.0±0.4	52.8±0.2	55.5±0.2	52.9±0.4	53.9±0.5	**54.9**±0.3
D→W	96.1±0.0	95.4±0.3	95.0±0.5	96.4±0.1	96.0±0.3	95.7±0.3	96.1±0.2	93.7±0.4	95.4±0.4	**96.9**±0.5

TABLE II

AVERAGE RECOGNITION ACCURACY (%) ON OFFICE-31 DATA SET WITH STANDARD SEMISUPERVISED ADAPTATION PROTOCOL [57], WHERE A = AMAZON, D = DSLR, AND W = WEBCAM

Config	CNN (S) [26]	CNN (T) [26]	CNN (S+T) [26]	DDC [45]	SDT [8]	DAN [7]	DLRC [18]	DTLC$_L$	DTLC$_N$
W→A	46.1±0.4	59.9±0.3	65.2±0.7	-	65.0±0.5	-	58.8±0.5	65.3±0.4	**66.9**±0.5
W→D	92.0±0.3	81.8±1.0	96.3±0.5	96.3±0.3	**97.6**±0.2	96.4±0.2	94.2±0.5	96.5±0.4	97.3±0.4
A→W	60.4±0.5	80.5±0.5	82.5±0.9	84.1±0.6	82.7±0.8	85.7±0.3	78.3±0.3	82.9±0.3	**86.4**±0.4
A→D	58.5±0.4	81.8±1.0	85.2±1.1	-	86.1±1.2	-	80.8±0.5	85.1±0.6	**86.8**±0.8
D→A	52.4±0.6	59.9±0.3	65.8±0.5	-	66.2±0.3	-	63.6±0.4	64.9±0.5	**67.3**±0.4
D→W	94.0±0.3	80.5±0.5	93.9±0.5	95.4±0.4	95.7±0.5	97.2±0.2	94.7±0.5	96.4±0.4	**97.9**±0.5

2) Deep CORAL [31] adapts CORAL loss [10] to couple the correlations between top layer's activations of a deep structure.
3) CNN [26] is a well-known framework to learn generic features. Here, we use the ImageNet to train the model.
4) Deep domain confusion (DDC) [45] is the first to incorporate domain confusion loss (linear-kernel MMD) to the top of AlexNet to mitigate the domain shift.
5) DAN [7] learns more transferable features by aligning three fully connected layers with multikernel MMD.
6) RevGrad [9] boosts domain adaptation using an adversarial loss (binary domain classifier) instead of MMD in DAN [7].
7) SDT [8] jointly seeks domain-invariant features and explores a soft label distribution matching to assist knowledge transfer. However, it needs some labeled target data during model training.
8) DHN [56] aims to seek informative hash coding by incorporating deep structure learning and domain alignment together.

For baseline comparisons, we adopt the standard procedures for model selection as described in their respective works. For our proposed model, we have two versions, i.e., linear one (DTLC$_L$) and nonlinear one (DTLC$_N$) and we conduct fivefold cross-valuation on the labeled data to choose candidate parameters. To be specific, we search parameters λ, α, and β in the range from 0.01 to 100. To suppress noisy predictions at the early stage of the whole training procedure, we set parameters α and β as zero within 1000 minibatch iterations. We implement our models using TensorFlow and train using Stochastic Gradient Descent with momentum. The initial learning rate is set as 10^{-3}, weight decay is 10^{-3}, and momentum is 0.9. Since the classifier is trained from scratch, we choose its learning rate 10 times than that of the lower layers. Moreover, we exploit minibatch with a batch size of 256, and generally 20 000 iterations are required and we observe the loss lower than 10^{-1}. We train on GPU with NVIDIA GeForce GTX TITAN X and generally it costs several hours to train the model, which is comparable with other deep transfer competitors. Specifically, we adopt the CNN structure of AlexNet [26]. Actually, our model can easily adopt other CNN structures, e.g., VGG, ResNet, and GoogleNet. Deeper CNN structures would improve the performance somehow. Since we are focusing on the specific layers, we only evaluate the AlexNet structure in this paper.

We follow standard evaluation protocols to validate all the algorithms in unsupervised/semisupervised domain adaptation settings. In unsupervised setting, we adopt the full-sampling protocol [7] that uses all labeled source samples while all unlabeled samples in the target domain are used to train the models. In semisupervised setting, three target samples are labeled. We adopt DeCAF$_7$ [59] features for shallow transfer methods and our conference version, DLRC [18], and original images for CNN-based deep transfer methods. We report the average classification performance in terms of top1 accuracy for each comparison over five random evaluations, also the standard deviation of the top1 accuracies by different evaluations on the same cross-domain task.

B. Comparison Results

The unsupervised/semisupervised adaptation results over the six learning tasks on Office-31 are listed in Tables I and II, the unsupervised adaptation results on 12 Office-10 + Caltech-10 cross-domain tasks are presented in Table III, and the unsupervised adaptation results on 12 Office + Home transfer tasks are reported in Table IV. Note that partial

TABLE III

AVERAGE RECOGNITION RATE (%)± STANDARD VARIATION OF 10 ALGORITHMS ON OFFICE + CALTECH-256 WITH FULL-SAMPLING PROTOCOL [7], WHERE A = AMAZON, D = DSLR, C = CALTECH-256, AND W = WEBCAM

Config	COREL [10]	CNN [26]	DDC[45]	DAN [7]	D-COREL [31]	DHN [56]	DLRC [18]	DTLC$_L$	DTLC$_N$
C→W	88.1±0.0	83.1±0.3	81.6±0.4	92.0±0.4	91.4±0.3	92.1±0.2	89.2±0.3	90.4±0.4	**92.3**±0.6
C→D	87.9±0.0	89.0±0.3	87.8±0.4	90.5±0.2	90.2±0.4	90.1±0.3	88.1±0.4	90.4±0.5	**91.4**±0.3
C→A	92.1±0.0	91.1±0.2	92.1±0.3	92.0±0.3	92.2±0.3	91.9±0.3	91.3±0.5	92.5±0.4	**93.2**±0.6
W→C	75.5±0.0	76.1±0.5	77.8±0.5	81.5±0.3	81.2±0.2	81.0±0.3	76.3±0.5	80.8±0.6	**82.1**±0.4
W→A	85.6±0.0	83.1±0.3	84.9±0.4	92.1±0.3	91.5±0.2	91.8±0.3	87.4±0.5	91.6±0.4	**93.3**±0.6
W→D	99.8±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	99.2±0.2	99.3±0.3	100.0±0.0	100.0±0.0
A→C	84.6±0.0	83.8±0.3	84.3±0.5	86.0±0.5	85.2±0.4	86.2±0.3	82.7±0.4	86.2±0.7	**87.2**±0.7
A→W	85.4±0.0	83.1±0.3	86.1±0.3	**93.8**±0.4	92.6±0.0	93.0±0.0	89.9±0.5	90.9±0.3	93.2±0.5
A→D	82.8±0.0	88.3±0.3	89.0±0.4	92.0±0.3	89.2±0.0	89.2±0.0	88.2±0.4	91.0±0.5	**93.1**±0.4
D→A	90.4±0.0	89.0±0.3	89.5±0.4	92.0±0.5	91.7±0.4	91.8±0.3	90.9±0.5	91.7±0.4	**92.8**±0.5
D→C	79.6±0.0	81.0±0.4	81.1±0.3	82.4±0.3	82.2±0.2	81.9±0.4	80.3±0.3	81.9±0.5	**82.9**±0.3
D→W	96.9±0.0	97.6±0.2	98.2±0.3	99.0±0.2	99.2±0.3	99.1±0.2	97.5±0.3	98.0±0.4	**99.3**±0.3

TABLE IV

RECOGNITION ACCURACIES (%) FOR DOMAIN ADAPTATION EXPERIMENTS ON THE OFFICE + HOME DATA SET, WHERE ART (AR), CLIPART (CL), PRODUCT (PR), AND REAL-WORLD (RW)

Config	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr
CORAL [10]	27.10	36.16	44.32	26.08	40.03	40.33	27.77	30.54	50.61	38.48	36.36	57.11
DAN [7]	30.66	42.17	54.13	32.83	47.59	49.78	29.07	34.05	56.70	43.58	38.25	62.73
RevGrad [9]	33.33	42.96	54.42	32.26	49.13	49.76	30.49	38.14	56.76	44.71	42.66	64.65
DHN [56]	31.64	40.75	51.73	34.69	51.93	52.79	29.91	39.63	60.71	44.99	**45.13**	62.54
D-COREL [31]	30.84	42.02	54.12	32.63	47.14	48.86	28.65	34.14	55.36	43.73	39.26	62.24
DLRC [18]	29.28	39.92	49.48	32.28	47.25	48.23	28.69	36.42	54.76	42.12	38.64	60.45
DTLC$_L$	32.52	42.63	54.12	34.42	49.93	51.86	30.23	39.43	59.76	44.13	41.26	63.05
DTLC$_N$	**34.49**	**43.63**	**55.28**	**36.14**	**52.74**	**53.16**	**31.59**	**40.55**	**61.43**	**45.64**	44.58	**65.92**

results are directly from [7]–[9], [31], and [56]. From the comparison results over different benchmarks, we can achieve the following observations.

First of all, traditional deep learning models (e.g., CNN) achieve comparable performance with conventional shallow transfer learning approaches over deep DeCAF$_7$ features as input (e.g., COREL). We could notice that the only difference between two lines of algorithms is COREL which could benefit from further feature adaptation over generic deep representations, while CNN could obtain gains over supervised fine-tuning on the labeled source data. This observation keeps pace with the recent discovery on deep neural networks, which are able to capture abstract features, but still leave domain mismatch there [43]. However, we could find that our DLRC still enhances the performance over deep generic features through multilayer strategy, although the improvement is limited.

Second, deep CNN-based transfer learning algorithms (e.g., DAN, RevGrad, SDT, D-COREL, and DHN) are capable of reducing the domain gap through domain alignment strategies and thus substantially outperform standard deep learning methods (CNN). This demonstrates that unifying domain alignment constraints over deep neural networks can assist unlabeled target learning. From the comparison of DDC and DAN, we could conclude that multilayer adaptation in the task-specific layers benefits a lot in knowledge transfer. RevGrad and SDT introduce the domain confusion into the deep structure learning, and therefore, they achieve comparable performance with DAN. Specifically, SDT incorporates the soft labels of target data during model training, which verifies such knowledge can benefit the classifier learning. However, they ignore the conditional distribution divergence across two domains. DHN adopts the same adaptation strategy with DAN but with hash coding as the last layer. Thus, DHN achieves the similar performance with DAN. However, DHN performs well in large-scale data set, i.e., Office + Home, which benefits from hash coding.

Besides, comparing with Office-31 and Office-10 + Caltech-10, Office + Home is more challenging, since it contains more categories, while other two have less categories. Another thing is four data sets in Office + Home have larger distribution divergence. Thus, all the algorithms cannot achieve promising performance. We could notice that our proposed model obtains better performance in most cases. Especially, in Office + Home data set, our model can achieve better performance than other comparisons, e.g., DAN.

Particularly, DTLC$_N$ consistently enhances the classification performance on some challenging cross-domain tasks, e.g., $A \rightarrow W$ and $A \rightarrow D$, in which the distribution divergences across source and target are substantially large. It also obtains comparable classification performance over some easy cross-domain tasks, e.g., $D \rightarrow W$ and $W \rightarrow D$, in which source and

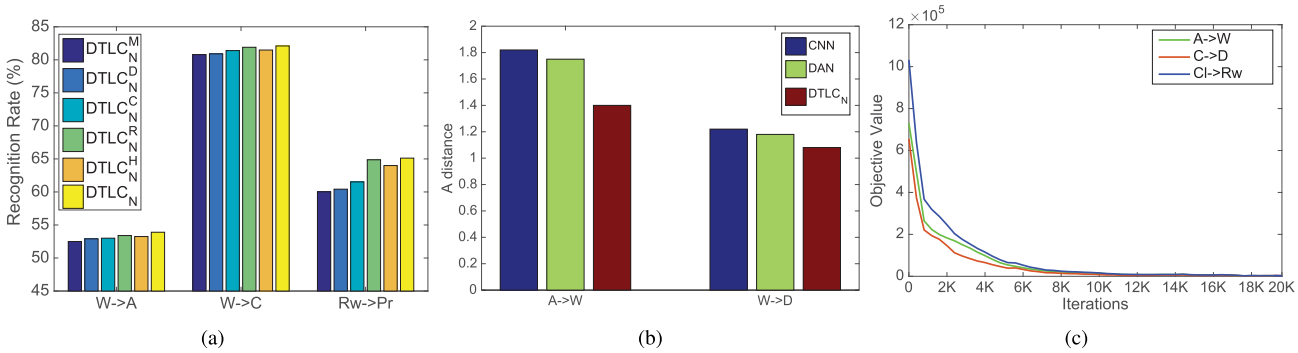


Fig. 2. (a) Recognition accuracy for different variants of our nonlinear model across three tasks. (b) $\mathcal{A}$-distance of three algorithms on two tasks in Office-31 using unsupervised domain adaptation protocol [7]. (c) Convergence curves of our model on three cases.

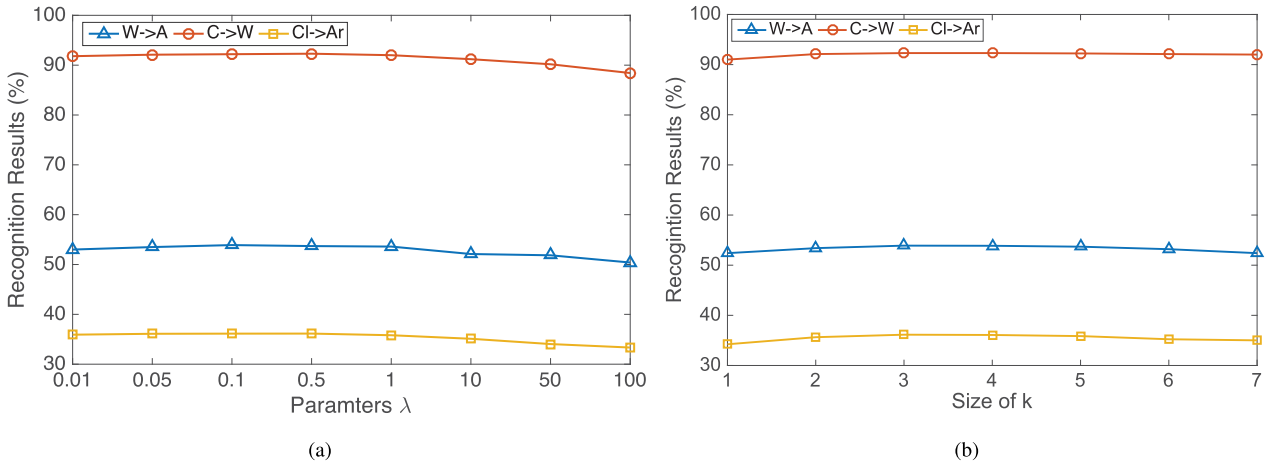


Fig. 3. (a) Influence of parameter λ and (b) impact of layer size with different values of k on three tasks across three cross-domain benchmarks for our nonlinear version, i.e., DTLC$_N$.

target only have the resolution difference. Size normalization used in CNN can further mitigate the resolution shift. The encouraging outputs suggest that our DTLC$_N$ is capable of seeking more domain invariant features and effective classifiers to facilitate the unlabeled target labeling. Different from conventional deep transfer learning approaches, our proposed architecture aims to mitigate the domain mismatch through multilayer dictionary decomposition, and furthermore, the low-rank constraint is also involved to uncover more classwise information. The novel designed classwise adaptation in the semisupervised scheme also contributes to reduce the conditional distribution divergence. We will discuss this phenomenon in Section IV-C.

C. Properties Analysis

In this section, we evaluate several properties of the proposed DTLC$_N$, i.e., convergence analysis, evaluation on variants, and so on.

1) Self-Evaluation: To dive deeper to the efficacy of our proposed model, we evaluate several variants of DTLC$_N$ as follows.

1) DTLC$_N^C$ means DTLC$_N$ without classwise adaptation, i.e., (6).
2) DTLC$_N^D$ represents DTLC$_N$ without adaptation on Z_k including both domainwise and classwise adaptations, i.e., $\alpha = 0$ in (7).
3) DTLC$_N^R$ denotes the DTLC$_N$ by removing the low-rank constraint on Z_k, i.e., $\beta = 0$.
4) DTLC$_N^M$ is the DTLC$_N$ by setting both α and β to 0.
5) DTLC$_N^H$ is the variant of DTLC$_N$ by replacing the soft label with the hard label of predicted target data, which is similar to the strategy of joint domain adaptation [48].

We experiment on three cases per data set using the unsupervised full-sampling protocol in [7]. From the result in Fig. 2(a), we could observe that the performance would drop when we remove the semisupervised classwise adaptation and further decrease by removing both two adaptations (domainwise and classwise terms). This verifies that our designed adaptation term can facilitate the knowledge transfer during model training. Besides, the rank constraint helps a little, seen from DTLC$_N^R$ and DTLC$_N$. We also find the performance of DTLC$_N^M$ hurts a lot by removing both the terms. Furthermore, the performance drops a lot when we remove all the knowledge transfer part and only remain the multilayer dictionary decomposition.

2) Convergence: There are two subproblems in our optimization, one is deep convolution neural networks and the other is parameters updating (i.e., probability label and classwise adaptation matrix). Therefore, it is challenging to theoretically verify its convergence. In this way, we follow researchers to empirically show the convergence of our model. Actually, we notice that our proposed model has

a good convergence property for all the cases and here we show some cross-domain tasks, i.e., {A, D} → W of Office-31, {A, C, W} → D of Office-10 + Caltech-10, and {Ar, Pr, Cl} → Rw of Office-Home data set in Fig. 2(c).

3) Distribution Discrepancy: The theory of domain adaptation [60] considers $\mathcal{A}$-distance [defined as $d_{\mathcal{A}} = 2(1 - 2\epsilon)$] as a way to measure the cross-domain discrepancy, where ϵ denotes the generalization error of a domain classifier trained over the two-class problem to identify source and target data (the similar to adversarial loss [8]). Fig. 2(b) reports $d_{\mathcal{A}}$ over two cross-domain tasks, i.e., $A \rightarrow W$ and $W \rightarrow D$ using the features obtained from CNN, DAN, and DTLC$_N$. From the results, we notice that $d_{\mathcal{A}}$ of our DTLC$_N$ is smaller than that of CNN and DAN, which indicates that DTLC$_N$ carries more transferable knowledge. Moreover, $d_{\mathcal{A}}$ of cross-domain task $W \rightarrow D$ is smaller than that of $A \rightarrow W$, since the mismatch of W and D is not large. That is why, $W \rightarrow D$ shows very promising recognition performance.

4) Parameter Analysis: There exist two key parameters in our model, e.g., λ to balance to rank term and reconstruction term, k layer size for the deep structure. Thus, we evaluate λ and the layer size across three data sets for our nonlinear version. For the influence of λ, we evaluate three tasks by grid searching from 0.01 to 100, whose results are shown in Fig. 3(a). We found that, usually, we could get better performance when λ is around 0.1. For the layer size, we experiment on different layer sizes k to see the impact of different values. Generally, we notice that $k = 3$ could generate better results from Fig. 3(b). A deeper structure may not contribute to the performance, which should depend on different architectures, e.g., AlexNet, VGG, and GoogLeNet. In this section, we only explore AlexNet architecture.

V. Conclusion

In this paper, we designed a novel DTLC framework, which jointed multilayer common dictionaries, low-rank coding, and CNN architecture into a unified optimization procedure. Specifically, common knowledge across two domains was exploited by multiple dictionaries in a layerwise fashion. Therefore, the learned low-rank coding could uncover more shared information of two domains to bridge the gap across two domains. Furthermore, the deep low-rank coding was guided with domain/classwise adaption terms, aiming to minimize the marginal and conditional distribution difference of two domains, so that the learned low-rank coding of two domains could be aligned well. Experimental results on three cross-domain benchmarks verified that our proposed model could outperform other comparisons by uncovering more shared knowledge of two domains.

References

[1] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2014.

[2] Z. Ding, M. Shao, and Y. Fu, "Robust multi-view representation: A unified perspective from multi-view learning to domain adaption," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 5434–5440.

[3] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 361–368.

[4] Z. Ding, M. Shao, and Y. Fu, "Latent low-rank transfer subspace learning for missing modality recognition," in *Proc. 28th AAAI Conf. Artif. Intell.* San Francisco, CA, USA: AAAI Press, 2014, pp. 1192–1198.

[5] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 56–63.

[6] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista, "Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3846–3854.

[7] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[8] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4068–4076.

[9] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[10] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. Assoc. Advancement Artif. Intell.*, 2016, pp. 2058–2065.

[11] S. Jiang, Z. Ding, and Y. Fu, "Deep low-rank sparse collective factorization for cross-domain recommendation," in *Proc. ACM Multimedia Conf.*, 2017, pp. 163–171.

[12] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1859–1867.

[13] Z. Ding, M. Shao, and Y. Fu, "Incomplete multisource transfer learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 310–323, Feb. 2018.

[14] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mar. 2018, pp. 8156–8164.

[15] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Webly-supervised fine-grained visual categorization via deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1100–1113, May 2018.

[16] H. Liu, M. Shao, Z. Ding, and Y. Fu, "Structure-preserved unsupervised domain adaptation," *IEEE Trans. Knowl. Data Eng.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8370901

[17] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, and P. S. Yu, "Transfer sparse coding for robust image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 407–414.

[18] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3453–3459.

[19] L. Niu, W. Li, D. Xu, and J. Cai, "An exemplar-based multi-view domain generalization framework for visual recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 259–272, Feb. 2018.

[20] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 21st AAAI Conf. Artif. Intell.*, 2008, pp. 677–682.

[21] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552–5562, Dec. 2016.

[22] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 37–52.

[23] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 692–699.

[24] J. Zheng, Z. Jiang, P. J. Phillips, and R. Chellappa, "Cross-view action recognition via a transferable dictionary pair," in *Proc. Brit. Mach. Vis. Conf.*, vol. 25, no. 6, 2012, pp. 1–11.

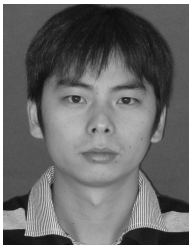
[25] S. Li, K. Li, and Y. Fu, "Self-taught low-rank coding for visual learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 645–656, Mar. 2018.

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

[27] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[28] Y. Yan, F. Nie, W. Li, C. Gao, Y. Yang, and D. Xu, "Image classification by cross-media active learning with privileged information," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2494–2502, Dec. 2016.

[29] 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, Aug. 2013.
[30] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 343–351.
[31] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. Workshops*. New York, NY, USA: Springer, 2016, pp. 443–450.
[32] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3801–3809.
[33] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.
[34] B. Quanz, J. Huan, and M. Mishra, "Knowledge transfer with low-quality data: A feature extraction issue," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 10, pp. 1789–1802, Oct. 2012.
[35] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 676–683.
[36] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.
[37] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.
[38] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.
[39] H. Venkateswara, P. Lade, J. Ye, and S. Panchanathan, "Coupled support vector machines for supervised domain adaptation," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1295–1298.
[40] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1114–1127, May 2018.
[41] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.
[42] Z. Ding, N. M. Nasrabadi, and Y. Fu, "Semi-supervised deep domain adaptation via coupled neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5214–5224, Nov. 2018.
[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
[44] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
[45] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. (2014). "Deep domain confusion: Maximizing for domain invariance." [Online]. Available: https://arxiv.org/abs/1412.3474
[46] B. Schölkopf, J. C. Platt, and T. Hoffman, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2008, pp. 513–520.
[47] T.-M. H. Hsu, W.-Y. Chen, C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with imbalanced cross-domain data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4121–4129.
[48] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.
[49] X. Cai, C. Ding, F. Nie, and H. Huang, "On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1124–1132.
[50] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.
[51] J. Malo, I. Epifanio, R. Navarro, and E. P. Simoncelli, "Nonlinear image representation for efficient perceptual coding," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 68–80, Jan. 2006.
[52] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
[53] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[54] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2691–2698.
[55] Y. Le Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.
[56] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. CVPR*, 2017, pp. 5018–5027.
[57] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.
[58] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[59] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 647–655.
[60] 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, nos. 1–2, pp. 151–175, May 2010.



Zhengming Ding (S'14–M'18) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA, in 2018.

Since 2018, he has been a Faculty Member with the Department of Computer, Information and Technology, Indiana University—Purdue University Indianapolis, Indianapolis, IN, USA. His current research interests include machine learning and computer vision, especially he devotes himself to develop scalable algorithms for challenging problems in transfer learning and deep learning scenario.

Dr. Ding was a recipient of the Best Paper Award, SPIE in 2016, the Best Paper Candidate, ACM MM in 2017, and the National Institute of Justice Fellowship from 2016 to 2018. He is currently an Associate Editor of the *Journal of Electronic Imaging*.

Yun Fu (S'07–M'08–SM'11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2004, respectively, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2007 and 2008, respectively.

Since 2012, he has been an Interdisciplinary Faculty Member with the College of Engineering and the College of Computer and Information Science, Northeastern University, Boston, MA, USA. He has authored or co-authored in leading journals, books/book chapters, and international conferences/workshops. His current research interests include machine learning, computational intelligence, big data mining, computer vision, pattern recognition, and cyber-physical systems.

Dr. Fu is a Fellow of IAPR, OSA, and SPIE; a Lifetime Senior Member of ACM; a Lifetime Member of AAAI and the Institute of Mathematical Statistics; and a member of ACM Future of Computing Academy, Global Young Academy, AAAS, and INNS. He was a Beckman Graduate Fellow from 2007 to 2008. He was a recipient of seven Prestigious Young Investigator Awards from NAE, ONR, ARO, IEEE, INNS, UIUC, and Grainger Foundation; nine Best Paper Awards from IEEE, IAPR, SPIE, and SIAM; and many major Industrial Research Awards from Google, Samsung, and Adobe. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEANING SYSTEMS. He serves as an Associate Editor, the Chair, a PC Member, and a reviewer for many top journals and international conferences/workshops.