Joint Learning of Multiple Latent Domains and Deep Representations for Domain Adaptation

Xinxiao Wu, *Member, IEEE*, Jin Chen, Feiwu Yu, Mingyu Yao, and Jiebo Luo, *Fellow, IEEE*

***Abstract*—In domain adaptation, the automatic discovery of multiple latent source domains has succeeded by capturing the intrinsic structure underlying the source data. Different from previous works that mainly rely on shallow models for domain discovery, we propose a novel unified framework based on deep neural networks to jointly address latent domain prediction from source data and deep representation learning from both source and target data. Within this framework, an iterative algorithm is proposed to alternate between 1) utilizing a new probabilistic hierarchical clustering method to separate the source domain into latent clusters and 2) training deep neural networks by using the domain membership as the supervision to learn deep representations. The key idea behind this joint learning framework is that good representations can help to improve the prediction accuracy of latent domains and, in turn, domain prediction results can provide useful supervisory information for feature learning. During the training of the deep model, a domain prediction loss, a domain confusion loss, and a task-specific classification loss are effectively integrated to enable the learned feature to distinguish between different latent source domains, transfer between source and target domains, and become semantically meaningful among different classes. Trained in an end-to-end fashion, our framework outperforms the state-of-the-art methods for latent domain discovery, as validated by extensive experiments on both object classification and human action-recognition tasks.**

***Index Terms*—Deep feature learning, domain adaptation, latent domain discovery, probabilistic hierarchical clustering.**

I. Introduction

MANY EXISTING studies have shown that domain adaptation methods can successfully solve the problem of dataset bias by reducing the domain distribution mismatch between different domains. With the ability to learn robust classifiers for a new and unexpected target environment, domain adaptation methods have been extensively studied in many visual-recognition tasks, such as object classification [1]–[10], object detection [11]–[13], and video-event recognition [14]–[16].

Most existing domain adaptation methods are either restricted to a single domain or treat each dataset as one domain. However, in practice, datasets for visual recognition usually are not deliberately collected with clearly identifiable domains due to many extraneous factors, such as intracategory appearance and pose variations, cluttered background, and various occlusions. Several existing works [17]–[19] reveal that a large amount of images or videos from one dataset may consist of multiple unknown domains and, thus, focus on automatically discovering multiple latent source domains for domain adaptation. All of these methods rely on first extracting the visual feature and then training a shallow model to exploit latent domains where feature learning and model training are independently handled.

This paper proposes a joint learning framework based on deep neural networks for simultaneously handling both latent domain discovery and representation learning tasks under a unified architecture, as shown in Fig. 1. In this framework, an iterative optimization algorithm is proposed to couple a clustering algorithm with deep neural networks, by alternating between utilizing the clustering method to discover latent source domains and training the deep networks with the domain membership as a supervision to learn the deep representations for all of the data. By combining latent domain discovery and representation learning into a unified architecture and optimizing it in an end-to-end manner, we can obtain not only more powerful representations but also more precise multiple latent source domains.

Specifically, a new probabilistic hierarchical clustering method is employed to divide the source domain into several latent clusters. Each source data is assigned a set of probabilities belonging to the multiple clusters. The deep model consists of a source convolutional neural network (CNN) and a target CNN with shared weights. In the training phase, the deep model is learned by optimizing over a domain prediction loss and a task-specific classification loss (e.g., object classification loss or action classification loss in our experiments). This enables the learned deep feature to be both latent domain distinguishing and semantically meaningful. Meanwhile, in order to reduce the data distribution discrepancy between the source and target domains, an additional domain confusion loss is introduced to enhance the feature transferability between different domains.

Manuscript received December 20, 2018; revised April 18, 2019; accepted June 3, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61673062. This paper was recommended by Associate Editor X. Li. *(Corresponding author: Xinxiao Wu.)*

X. Wu, J. Chen, F. Yu, and M. Yao are with the Beijing Laboratory of Intelligent Information Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: wuxinxiao@bit.edu.cn; chen_jin@bit.edu.cn; yufeiwu@bit.edu.cn; yaomingyu@bit.edu.cn).

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14611 USA (e-mail: jiebo.luo@cs.rochester.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/TCYB.2019.2921559

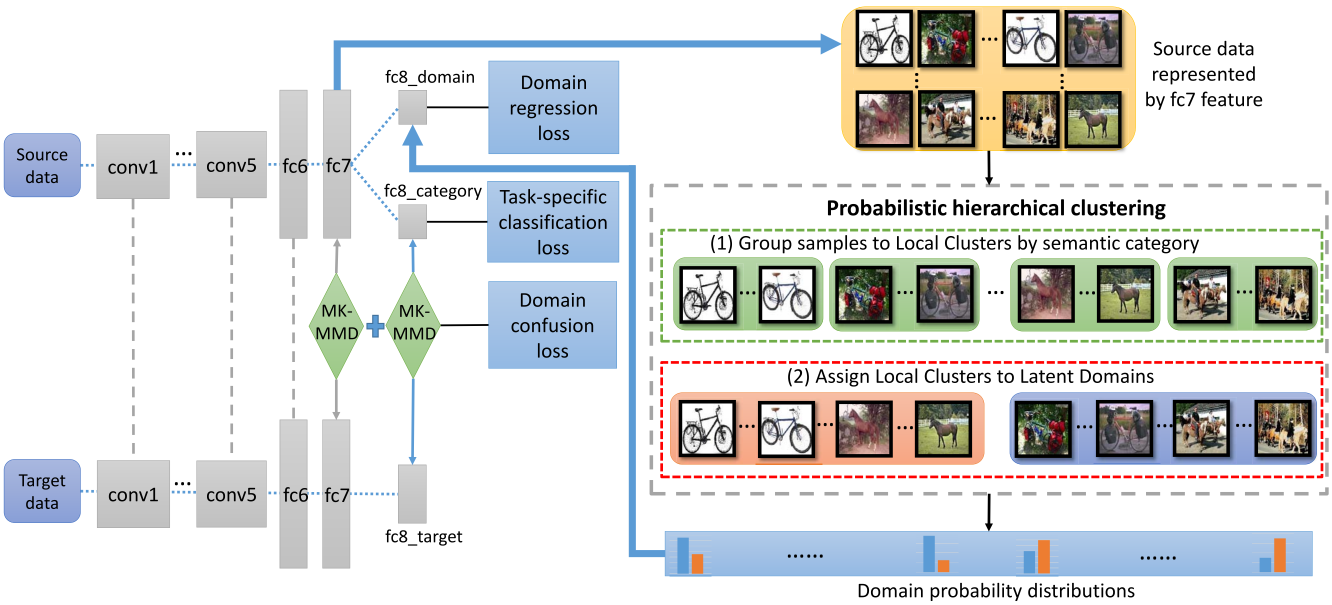

Source data
conv1
conv5
fc6
fc7
fc8_domain
Domain regression loss
fc8_category
Task-specific classification loss
MK-MMD
MK-MMD
Domain confusion loss
Target data
conv1
conv5
fc6
fc7
fc8_target
Source data represented by fc7 feature
Probabilistic hierarchical clustering
(1) Group samples to Local Clusters by semantic category
(2) Assign Local Clusters to Latent Domains
Domain probability distributions

Fig. 1. Overview of the joint learning framework of multiple latent domains and deep representation for domain adaptation.

Overall, the main contributions are as follows.

1) We propose a novel deep neural-network-based framework for domain adaptation, which jointly learns multiple latent source domains and deep representations for both source and target data.
2) We formulate joint learning as a new unified end-to-end training process that alternates between hierarchical clustering and representation learning by deep neural networks.
3) Extensive experiments on both object classification and action-recognition tasks show that the proposed framework outperforms the existing methods in discovering latent domains. In addition, our framework also achieves better results than other deep adaptation methods by effectively transferring the adapted classifiers learned on the discovered multiple latent source domains to the target domain.

The organization of the rest of this paper is given as follows. In Section II, we summarize the related works of latent domains discovery and deep domain adaptation. Section III describes the proposed method for domain adaptation based on identifying multiple latent source domains, including probabilistic hierarchical clustering, network architecture, and iterative algorithms. Section IV elaborates on the experimental result and analysis. The conclusion is made in Section V.

II. Related Work

A. Latent Domains Discovery

In terms of discovering multiple latent source domains, several papers [17]–[20] are closely related to this paper. Hoffman *et al.* [17] proposed a clustering-based method to discover the latent domains by deriving a hierarchically constrained assignment algorithm. Gong *et al.* [18] proposed a nonparametric method to automatically partition the source data into multiple latent domains which can simultaneously maximize the distinctiveness and learnability of the extracted domains. Xiong *et al.* [20] first proposed a novel local subspace representation for each data with the help of the relationship to its neighbors of the same category and then introduced the mutual information between subspace representation and domain identification as well as the prior distribution of the category in each domain for domain adaptation. Different from the above methods [17], [18], [20], Li *et al.* [19] proposed an exemplar-SVMs-based approach for both domain adaptation and generalization by explicitly exploiting the intrinsic structure of positive samples from multiple latent domains without dividing the training samples into multiple domains (or clusters). Different from these shallow methods, our deep model-based method simultaneously addresses domain discovery and feature learning in a unified framework, significantly improving domain adaptation performance.

A recent method [21] based on the deep neural network is most related to our method, which introduces novel domain alignment layers and a side branch into CNN to discover latent domains for boosting domain adaptation. Compared with this paper, our method is based on an iterative procedure between traditional clustering and deep model training with more flexibility, and any existing clustering method and CNN model can be easily embedded into our architecture. In addition, our method can be readily applied to any multiple source domain adaptation methods.

B. Deep Domain Adaptation

From the perspective of domain adaptation enabled by deep neural networks, this paper is more related to [5] and [22]–[29]. Tzeng *et al.* [22] introduced an adaptation layer into a traditional CNN architecture and design an additional domain confusion loss in the objective function for the purpose of reducing the data bias between different domains.

Long *et al.* [23] proposed a new deep adaptation network that reduces the domain discrepancy in higher task-specific layers using an optimal multikernel selection method for mean embedding matching. Ganin and Lempitsky [25] introduced a simple new gradient reversal layer in deep architectures for unsupervised domain adaptation. Through the standard backpropagation training progress, the learned deep features perform invariant to the shift between different domains and discriminative for the task on the source domain. In [26], a new deep domain adaptation approach is proposed to jointly learn adaptive classifiers and transferable features using labeled source domain data as well as unlabeled target domain data, under the assumption that the difference between the source and target classifiers is represented by a residual function. In [27], a novel architecture called domain separation networks is proposed to learn domain-invariant representations by exploiting both private and shared components of feature representations for both source and target domains. Tzeng *et al.* [28] proposed a unified deep framework called adversarial discriminative domain adaptation to handle unsupervised domain adaptation by combining discriminative modeling, untied weight sharing, and generative adversarial network (GAN) loss. In [29], a joint adversarial discriminative approach that leverages unsupervised data is proposed to transfer the information of the target distribution to the learned joint feature space using a generator–discriminator pair. Long *et al.* [24] proposed a joint distribution discrepancy of features and labels to measure the domain distance between different domains, and learn a set of joint adaptation networks by minimizing the joint distribution discrepancy. In contrast to these deep domain adaptation methods that are restricted to a single training domain, our proposed method is capable of automatically discovering the multiple hidden domains by coupling a deep model with a probabilistic hierarchical clustering strategy.

III. Deep Neural Networks With Probabilistic Hierarchical Clustering

Primarily focusing on an unsupervised scenario where the labeled source data are accompanied by unlabeled target data in the training phase, our goal is to simultaneously calculate the latent domain probabilities for the source data and learn the visual features for both source and target domains. Motivated by the effectiveness of deep neural networks in a variety of visual tasks, we employ a deep CNN for joint latent domain discovery and feature learning. Since automatically discovering the probabilities of a sample from multiple latent source domains is an unsupervised learning problem that cannot be easily handled by CNN, we couple the CNN model with a probabilistic hierarchical clustering method to convert unsupervised learning into supervised learning. It alternates between clustering the source data to predict the domain probabilities and training CNN models by using the predicted domain membership to supervise learning the deep representations of both source and target domains.

Given a source domain D_s and a target domain D_t, let $X_s = \{x^s_i|^{n_s}_{i=1}\}$ represent the labeled source data of n_s samples, where x^s_i is the feature of the ith source sample and $Y_s = \{y^s_i|^{n_s}_{i=1}\}$ are the corresponding category labels of X_s, where $y^s_i \in \{1, 2, \ldots, C\}$ is the label of x^s_i. The unlabeled target data of n_t samples is represented by $X_t = \{x^t_i|^{n_t}_{i=1}\}$, where x^t_i is the feature of the ith target sample. Suppose there are K latent domains to be discovered, $p^k_i \in [0, 1]$ represents the probability that the ith source sample x^s_i is assigned to the kth latent domain, where $k \in \{1, \ldots, K\}$. For each $x^s_i \in X_s$, let $P_i = \{p^k_i|^K_{k=1}\}$ denote the probability distribution of latent domain assignments for x^s_i with the constraint of $\sum_{k=1}^K p^k_i = 1$.

A. Probabilistic Hierarchical Clustering

Discovering latent domains in the source domain is difficult since the data are naturally separated according to semantic categories in many cases. Thus, the data tend to be clustered according to the category labels via traditional clustering methods, such as k-means, which contradicts the fact that samples within the same category are likely to come from multiple domains. Our clustering method can solve this problem by simultaneously considering the category and domain information. Different from [17], which constrains each sample to be assigned to one latent domain, our method calculates a set of probabilities for each sample that assigns it to multiple latent domains. Our method has two stages. In the first stage, the Gaussian mixture model is employed to model the probability distribution of source domain samples in which each Gaussian component corresponds to one local cluster. In the second stage, an EM-style algorithm is used to determine the division of the source domain by assigning the local clusters to the latent domains.

Let C and K denote the numbers of semantic categories and latent domains to be discovered, respectively. Given the source domain data samples $X_s = \{x^s_i|^{n_s}_{i=1}\}$ and the corresponding category labels $Y_s = \{y^s_i|^{n_s}_{i=1}\}$ with $y^s_i \in \{1, 2, \ldots C\}$, the data samples within the same category are grouped into M local clusters, and then the total number of local clusters is $J = M \cdot C$. For each local cluster, a single Gaussian component is utilized to model its probability distribution, denoted by the parameter set $\{\pi_{j^c_m}, \mu_{j^c_m}, \delta_{j^c_m}\}$, where $\pi_{j^c_m}$, $\mu_{j^c_m}$, and $\delta_{j^c_m}$ are the weight, mean, and variance of the mth single Gaussian model in the cth category, respectively. Let j^c_m represent the mth local cluster of the cth category with $m \in \{1, \ldots, M\}$ and $c \in \{1, \ldots, C\}$, and $p^L_{i,j^c_m} \in [0, 1]$ represents the probability that the data sample $x^{s,c}_i$ of the cth category is assigned to the local cluster j^c_m. We alternately optimize p^L_{i,j^c_m} and $\{\pi_{j^c_m}, \mu_{j^c_m}, \delta_{j^c_m}\}$. Specifically, with the fixed parameter set $\{\pi_{j^c_m}, \mu_{j^c_m}, \delta_{j^c_m}\}$, p^L_{i,j^c_m} is calculated by

$$p^L_{i,j^c_m} = \frac{\pi_{j^c_m} N\left(x^{s,c}_i | \mu_{j^c_m}, \delta_{j^c_m}\right)}{\sum_{q=1}^{M} \pi_{j^c_q} N\left(x^{s,c}_i | \mu_{j^c_q}, \delta_{j^c_q}\right)} \tag{1}$$

where $N(\cdot)$ denotes the single Gaussian model. With the fixed p^L_{i,j^c_m}, the parameters $\{\pi_{j^c_m}, \mu_{j^c_m}, \delta_{j^c_m}\}$ are formulated by

$$\pi_{j^c_m} = \frac{\sum_{i=1}^{n_c} p^L_{i,j^c_m}}{n_c}, \quad \mu_{j^c_m} = \frac{\sum_{i=1}^{n_c} p^L_{i,j^c_m} \cdot x^{s,c}_i}{\sum_{i=1}^{n_c} p^L_{i,j^c_m}} \tag{2}$$

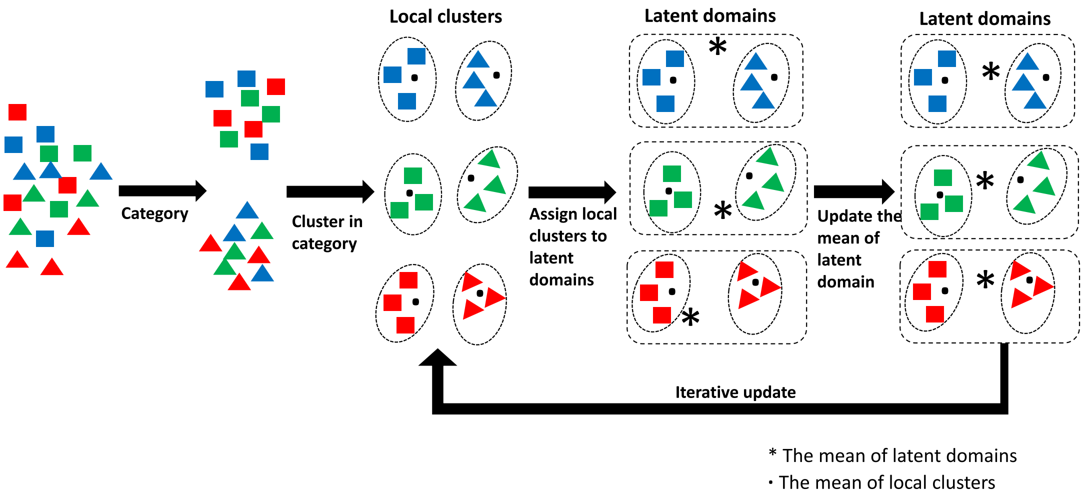

Local clusters
Latent domains
Latent domains
Category
Cluster in category
Assign local clusters to latent domains
Update the mean of latent domain
Iterative update
* The mean of latent domains
· The mean of local clusters

Fig. 2. Process of probabilistic hierarchical clustering. There are three latent domains and two categories. For each category, the data are divided into three local clusters. Different shapes represent different categories. Different colors represent different latent domains. The solid circle and star represent the mean of a local cluster and the mean of a latent domain, respectively. (This figure is best viewed in color.)

$$\delta_{j_m^c} = \frac{\sum_{i=1}^{n_c} p^L_{i,j_m^c}\left(x_i^{s,c} - \mu_{j_m^c}\right)\left(x_i^{s,c} - \mu_{j_m^c}\right)^T}{\sum_{i=1}^{n_c} p^L_{i,j_m^c}} \tag{3}$$

where n_c is the number of samples in the cth category.

After modeling the local clusters using a Gaussian mixture model, an EM-style iterative algorithm is adopted to estimate the optimal latent domains by assigning the local clusters to the latent domains. With the constraint that each latent domain must have only one local cluster from each category, the number of latent domains, therefore, is equal to the number of local clusters in each category, that is, $M = K$. Then, for each category, the number of all possible assignments of local clusters to latent domains is $A_K = K!$ Let $W_c \in \mathbb{Z}^{A_K \times K}$ represent all assignments for the cth category. Then, if $W_c(\sigma, k) = m$, it means that the mth local cluster in the cth category is assigned to the kth latent domain under the σth assignment, where σ represents the index of the assignment with $\sigma \in \{1, \dots, A_K\}$. Each latent domain is represented by its mean and we use u_k to indicate the mean of the kth latent domain with $k \in \{1, 2, \dots, K\}$. Let $p^G_{j_m^c,k}$ indicate the probability that the local cluster j_m^c is assigned to the kth latent domain. First, randomly initialize u_k and then alternately update $p^G_{j_m^c,k}$ and u_k. With the fixed u_k, $p^G_{j_m^c,k}$ is calculated by

$$p^G_{j_m^c,k} = \frac{\sum_{\sigma=1}^{A_k} \mathbb{1}_{W_c(\sigma,k)=m} \frac{1}{d(\mu_{j^c}, u|\sigma)}}{\sum_{\sigma=1}^{A_k} \frac{1}{d(\mu_{j^c}, u|\sigma)}} \tag{4}$$

where $\mu_{j^c} = \{\mu_{j_1^c}, \mu_{j_2^c}, \dots, \mu_{j_M^c}\}$ represents the set of means of local clusters in the cth category and $u = \{u_1, u_2, \dots, u_K\}$ represents the set of means of latent domains. $\mathbb{1}_{W_c(\sigma,k)=m}$ is an indicator function which represents that if $W_c(\delta, k) = m$, the value of $\mathbb{1}_{W_c(\sigma,k)=m}$ is 1 and otherwise is 0. $d(\mu_{j^c}, u|\sigma)$ denotes the Euclidean distance between the set of means of local clusters in the cth category and the set of means of latent domains under the σth assignment, given by

$$d\left(\mu_{j^c}, u|\sigma\right) = \sum_{k=1}^{K} \left\| \mu_{j^c_{W_c(\sigma,k)}} - u_k \right\|_2 \tag{5}$$

where $\| \cdot \|_2$ denotes the l_2-norm. With the fixed $p^G_{j_m^c,k}$, u_k is updated by the weighted average of all means of the local clusters from all categories

$$u_k = \frac{\sum_{c=1}^{C} \sum_{m=1}^{M} p^G_{j_m^c,k} \cdot \mu_{j_m^c}}{M \cdot C} \tag{6}$$

where $\mu_{j_m^c}$ represents the mean of the mth local cluster in the cth category, $p^G_{j_m^c,k}$ is the weight of $\mu_{j_m^c}$ and actually indicates the probability of assigning the mth local cluster in the cth category to the kth latent domain, and $M \cdot C$ is the number of all the local clusters. Finally, the probability p_i^k that x_i^s belongs to the kth latent domain is computed by

$$p_i^k = \sum_{m=1}^{M} p^L_{i,j_m^c} \cdot p^G_{j_m^c,k}. \tag{7}$$

The probability distribution of the latent domain assignment for x_i^s is represented by $P_i = \{p_i^k|_{k=1}^K\}$, where $\sum_{k=1}^K p_i^k = 1$. Fig. 2 simply demonstrates the process of probabilistic hierarchical clustering.

B. Network Architecture

As shown in Fig. 1, our deep architecture consists of a source CNN and a target CNN with shared weights. We extend the AlexNet architecture [30], which is a proven powerful model when adapting to novel tasks. It contains eight learned layers, including five convolutional layers (conv1–conv5) and three fully connected layers (fc6–fc8).

Given the training source data $X_s = \{x_i^s|_{i=1}^{n_s}\}$ with the corresponding latent domain probabilities $P = \{P_i|_{i=1}^{n_s}\}$, we design a latent domain prediction loss $L_r(X_s, P)$ on the fc8 layer to enable the learned deep representations distinguishable between different latent domains, formulated by

$$L_r(X_s, P) = \frac{1}{n_s} \sum_{i=1}^{n_s} E\left(\theta_r\left(x_i^s\right), P_i\right) \tag{8}$$

where $E(\cdot)$ is an Euclidean loss function and $\theta_r(x_i^s)$ is a set of conditional probabilities that assign x_i^s to multiple latent domains in the CNN.

Given the training source data X_s with the corresponding task-specific (e.g., object classification or action recognition in our experiments) category labels Y_s, a task-specific classification loss $L_c(X_s, Y_s)$ is added on the fc8 layer to make the representations semantically meaningful across different categories, defined as

$$L_c(X_s, Y_s) = \frac{1}{n_s}\sum_{i=1}^{n_s} F\big(\theta_c\big(x_i^s\big), y_i^s\big) \tag{9}$$

where $F(\cdot)$ represents the cross-entropy loss and $\theta_c(x_i^s)$ is the condition probability of classifying the sample x_i^s into the task-specific category label y_i^s.

In standard CNNs, deep representations change from general to specific, and eventually its transferability difficulty increases with the discrepancy of domains. Therefore, it is particularly difficult to transfer representations in the higher layers fc6–fc8. Since the fc layers cannot be directly transferred from the source to the target by just fine-tuning the original source CNN with limited target training data, we introduce a domain confusion loss and place it on top of the fc7 and fc8 layers to make the representation invariant to the source and target domains. This domain confusion loss $L_d(X_s, X_t)$ is represented by the maximum mean discrepancy (MMD) [31], which measures the distance between the source domain and the target domain based on kernels, defined by

$$\begin{aligned} L_d(X_s, X_t) &= \left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\phi\big(x_i^s\big) - \frac{1}{n_t}\sum_{i=1}^{n_t}\phi\big(x_i^t\big)\right\|^2 \\ &= \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} k\Big(x_i^s, x_j^s\Big) + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} k\Big(x_i^t, x_j^t\Big) \\ &\quad - \frac{2}{n_s \cdot n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} k\Big(x_i^s, x_j^t\Big) \end{aligned} \tag{10}$$

where $\phi(\cdot)$ defines a representation operating on n_s labeled source data $x_i^s \in X_s$ and n_t unlabeled target data $x_i^t \in X_t$. A characteristic kernel $k(x_i^s, x_i^t) = \langle \phi(x_i^s), \phi(x_i^t)\rangle$ is defined as a linear combination of m PSD kernels $\{k_u\}$

$$\mathbf{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \quad \forall u \right\} \tag{11}$$

where β_u is the weight for the uth kernel. In our experiments, we use an RBF kernel $e^{-(1/2\gamma)\|x_i^s - x_i^t\|^2}$ with the bandwidth γ which is set to the median pairwise distances on the training data. The constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that the derived multikernel k is characteristic and learned by the strategy in DAN [23]. We vary the bandwidth γ_u between $2^{-8}\gamma$ and $2^8\gamma$ with a multiplicative step-size of $2^{1/2}$. Due to the distribution change of the shared features during learning, it is beneficial to have a large range of kernels.

By effectively combining the latent domain prediction loss $L_r(X_s, P)$, the classification loss $L_c(X_s, Y_s)$, and the domain confusion loss $L_d(X_s, X_t)$, the final optimization problem to train the deep model can be given by

$$\min_{\Theta}(L_r(X_s, P) + L_c(X_s, Y_s) + L_d(X_s, X_t)) \tag{12}$$

where $\Theta = \{(W^l, b^l)|_{l=1}^{L}\}$ denotes the parameter set of the CNN model with the weights W^l and bias b^l of the lth fc layer. Since the source CNN and the target CNN share the same network architecture with the same weights, the learned features of both source and target data become distinguishable regarding different latent source domains, discriminative regarding different categories, and transferable across different domains with the help of $L_r(X_s, P)$, $L_c(X_s, Y_s)$, and $L_d(X_s, X_t)$.

C. Iterative Algorithm

We first extract the initial features X_s of the source domain D_s using the output of fc7 layer in the AlexNet [30]. We then calculate the domain probability distributions P of D_s based on the current X_s using the clustering method described in Section III-A. Next, we train the CNN model described in Section III-B with the predicted latent domain probabilities P and the given task-specific category labels Y_s as a supervision to update the source features X_s and the target features X_t. The above steps are repeated until the learned feature converges or the maximum number of iterations is reached. The detailed iterative optimization algorithm is summarized in Algorithm 1.

IV. Experiments

Our framework is validated on both visual object classification and human action-recognition tasks. First, we introduce the datasets with an evaluation strategy, and then describe the experimental settings. After that, we report the recognition accuracies of adapted classifiers from the source domain to the target domain by identifying multiple latent source domains and compare our method with other related methods.

A. Datasets

For object recognition, we use images from the datatsets of ImageNet (I) [30], Caltech-256 (C) [32], Pascal VOC (P) [33], and Bing (B) [34]. A total of 12 common categories among the four datasets are adopted in our experiment, including "airplane," "bike," "bird," "boat," "bottle," "bus," "car," "dog," "horse," "monitor," "motorbike," and "people." For each category in each dataset, about 100 images are randomly selected to construct the training and test data. The target domain is composed of unlabeled images from one dataset while the source domain is constructed by the labeled images from the remaining three datasets. We permute all of the domain combinations and set up four domain adaptation tasks: 1) CPB → I; 2) IPB → C; 3) ICB → P; and 4) ICP → B. For each category in the target domain, 30 images are randomly selected to construct the dataset for training, and the remaining images are used for testing.

For human action recognition, the source domain is composed of the images from the Stanford40 dataset [35], and the target domain consists of the videos from the UCF101 dataset [36]. There are a total of 12 common action classes

Algorithm 1 Deep Neural Networks With Constrained Clustering for Latent Domain Discovery

```
Input: The Source domain $D_s$, the task-specific category labels $Y_s$ of $D_s$, the target domain $D_t$, and the number of domains $K$.
Output: Latent domain probability distributions $P$ of $D_s$, the deep features $X_s$ and $X_t$ of $D_s$ and $D_t$, respectively.
Extract the initial features $X_s$ and $X_t$ from the fc7 layers of the source CNN and target CNN, respectively.
repeat
    Compute the probabilities $p^L_{i,j^c_m}$, $i \in \{1, 2, ..., n_s\}$ that the source data $X_s$ are assigned to the local clusters $j^c_m$, $m \in \{1, 2, ..., M\}$, $c \in \{1, 2, ..., C\}$ by Eq.(1).
    Randomly initialize the means of latent domains $u_k, k \in \{1, 2, ..., K\}$.
    repeat
        Compute the probabilities $p^G_{j^c_m,k}$ that the local clusters $j^c_m$ are assigned to the latent domains by Eq.(4).
        Update the means of latent domain $u_k$ by Eq.(6).
    until Converge
    Compute the latent domain probability distributions $P = \{P_i|_{i=1}^{n_s}\}$ where $P_i = \{p_i^k|_{k=1}^{K}\}$ using $p^L_{i,j^c_m}$ and $p^G_{j^c_m,k}$ by Eq.(7).
    Train the source and target CNN models with $D_s$, $D_t$, $Y_s$ and $P$ by Eq.(12).
    Extract the new features $\hat{X}_s$ and $\hat{X}_t$ from the fc7 layers of the updated source CNN and target CNN, respectively.
    Update $X_s$ and $X_t$ by $\hat{X}_s \Rightarrow X_s$ and $\hat{X}_t \Rightarrow X_t$.
until Converge
```

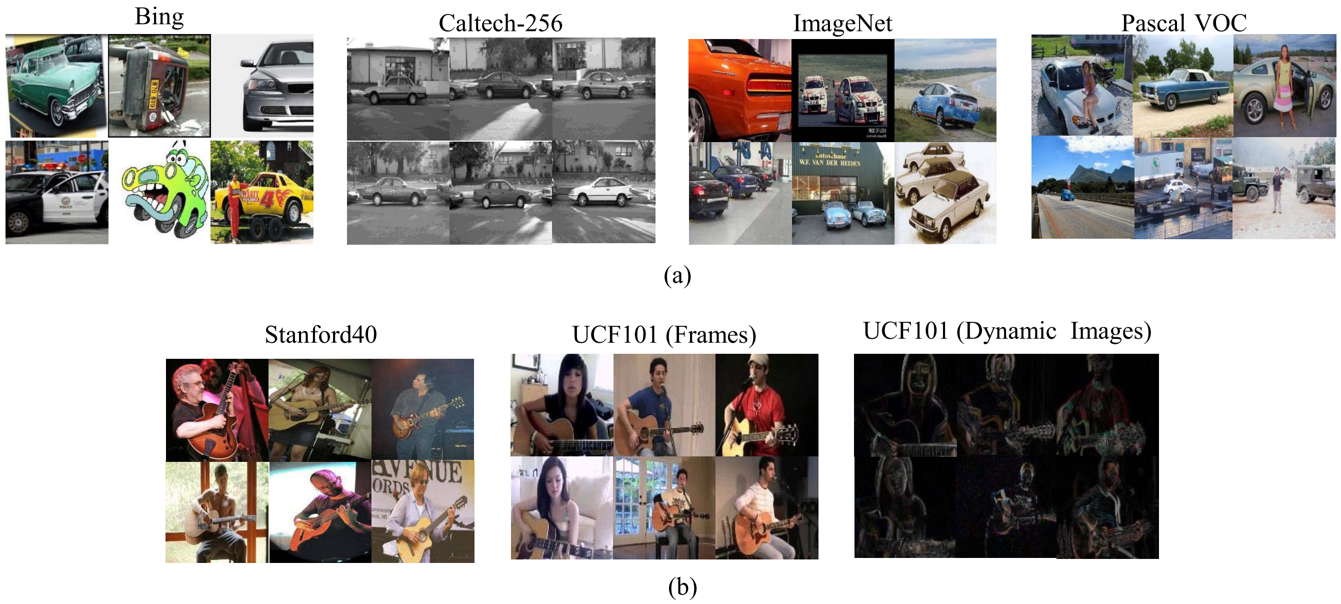


Fig. 3. Illustration of several image examples for (a) object classification and (b) action recognition.

among the two datasets, including "brushing teeth," "cleaning floor," "climbing," "cutting vegetables," "playing guitar," "playing violin," "biking," " horse riding," "rowing," "shooting," "walking with dog," and "writing on a board." For each class in the target domain, 50 videos are randomly chosen for training, and the remaining samples are taken for testing. For the videos, the dynamic image [37] for each sample is extracted to capture the motion formation of actions and used as the input to the CNN of the target domain. Some image examples for object classification and action recognition are illustrated in Fig. 3.

To evaluate the effectiveness of our approach, we focus on investigating whether the recognition performance on the target domain can be improved by automatically discovering multiple latent source domains for domain adaptation. Specifically, we employ several multiple source-domain adaptation methods (i.e., DSM [38], DAM [39], M-GFK, M-LRSR, M-TJM, and MDAN [42]) to adapt the classifiers trained on the identified latent domains to the target domain and use the recognition accuracy on the target domain to validate the performance of the latent domains. DSM, DAM, and MDAN are multiple-domain adaptation methods and GFK [1], LRSR [40], and TJM [41] are single-domain adaptation methods.

1) DSM selects the most relevant source domain to adapt to the target domain, which enforces the target classifier to share the same decision values with the corresponding source classifiers.
2) DAM proposes a new framework for learning a target classifier by using a set of classifiers pretrained on labeled samples from multiple source domains.

3) MDAN proposes a novel multisource-domain adversarial network.
4) GFK proposes a geodesic flow kernel in order to leverage low-dimensional feature structures.
5) LRSR transforms both the source and target data to a shared feature space, in which source samples can be effectively combined to represent each sample in the target domain.
6) TJM learns a feature space to minimize the domain distance by a newly designed transfer joint matching algorithm and reweights the source instances irrelevant to the target instances with less importance.

For multiple domain adaptation, we extend the single-domain adaptation methods of GFK, LRSR, and TJM into the multidomain version, called M-GFK, M-LRSR, and M-TJM, respectively, by averaging the decision values obtained from the classifiers trained on all source domains.

To evaluate the effectiveness of utilizing the deep neural networks for latent domain discovery, we compare our deep model with several traditional shallow models: Latent [17], Reshape [18], and LRE-SVMs [19]. For these three methods, the output from the fc7 layer of the AlexNet [30] is used as the visual representation. For the Latent method, we set the threshold of terminating iteration to 0.001 and the maximum iteration number to 5. For the Reshape method, we use Gaussian kernels and the kernel bandwidth is set to be twice the median distances of all pairwise data points. For the LRE-SVMs method, we set the relaxation factor C in SVM to 0.001 and the tradeoff parameter λ to 1.

To validate the advantage of exploiting multiple latent source domains for domain adaption in the deep architecture, our method is also compared with a variety of deep domain adaptation methods: CNN [30], DDC [22], DAN [23], RevGrad [25], RTN [26], JAN [24], AutoDIAL [43], WDGRL [44], and mDA [21].

1) CNN is a powerful deep network for learning transferable and discriminative features.
2) DDC adds an adaption layer between the fc7 and fc8 layers to maximize domain invariance and designs an additional domain confusion loss in the objective function to reduce the data bias between different domains.
3) DAN introduces multiple kernel learning to match the mean embeddings of the source and target data distributions for reducing the domain discrepancy in higher task-specific layers of deep neural networks.
4) RevGrad introduces a simple new gradient reversal layer in deep architectures to learn the deep features that are discriminative for the task on the source domain and invariant with respect to the shift between the domains.
5) RTN jointly learns adaptive classifiers and transferable features from labeled source data and unlabeled target data with the assumption that the main difference between the source and target classifiers is formulating a residual function.
6) JAN reduces the domain shift by aligning the joint distributions of multiple domain-specific layers, which is implemented by jointly maximizing mean discrepancies of feature distributions of these layers.
7) AutoDIAL resorts to aligning both source and target distributions to a reference one, which is implemented by adding new domain alignment layers to a deep neural network to handle the domain discrepancy.
8) WDGRL takes advantage of the gradient property of Wasserstein distance to reduce the domain discrepancy, and the transferability is guaranteed by the generalization bound.
9) mDA utilizes a new deep neural network with an additional branch to compute a set of probabilities for each sample that assigns it to multiple latent domains and designs multidomain domain adaptation layers to handle the domain shift.

All of these deep methods are implemented under the Caffe framework [45], and fine-tuned from Caffe-trained models of Alexnet which are pretrained on the ImageNet. We employ the mini-batch stochastic gradient descent (SGD) to train the deep networks and set the momentum to 0.9. The learning rate of all convolutional and pooling layers is set to 0.0001, as these layers are fine-tuned from the Alexnet model. We set the learning rate of the domain classifier and the task-specific classifier to 0.001 to train them from scratch. All of the experiments are carried out on a single GeForce GTX Titan X GPU. At the feature extraction time, all models run well in 1 s on this GPU.

B. Results

1) Comparison With Latent Domain Discovery Methods: Table I reports the domain adaptation results of our method and traditional methods of latent domain discovery for both object classification and action-recognition tasks. We can make the following observations.

1) Our method achieves the best performance for all multiple source-domain adaptation methods on both datasets, which explicitly demonstrates the effectiveness of performing latent domain identification and feature learning in a unified deep architecture.
2) For the methods of Latent, Reshape, and Ours, the final result is further influenced by the performance of the multiple source-domain adaptation method. For all of the latent domain discovery methods, the methods of DSM and DAM generally outperform those of M-GFK, M-LRSR, M-TJM, and MDAN. One possible reason is because DSM and DAM are able to adaptively transfer the knowledge by automatically learning different weights of different latent source domains while M-GFK, M-LRSR, M-TJM, and MDAN equally treat each latent domain during the transfer.
3) The LRE-SVMs method does not explicitly divide the source data into multiple domains and cannot be applied to the methods of multiple source-domain adaptation, which performs worse than our method on all tasks.
4) In comparison with object classification, the accuracies on the action-recognition task are much lower for all methods since the dynamic image representation of videos in the target domain (UCF101 dataset) is very different from the image from the source domain

TABLE I
ACCURACIES (%) OF DIFFERENT METHODS OF DISCOVERING LATENT DOMAINS ON THE OBJECT DATASET AND THE ACTION DATASET

	Object Classification					Action Recognition
Source domain	**CPB**	**IPB**	**ICB**	**ICP**	Avg	**Stanford40**
Target domain	**I**	**C**	**P**	**B**		**UCF101**
Latent[17]+DSM[38]	83.9	92.5	65.7	61.2	75.8	18.2
Latent[17]+DAM[39]	83.1	90.6	66.4	58.6	74.7	17.1
Latent[17]+M-GFK[1]	72.3	71.3	57.0	51.4	63.0	20.3
Latent[17]+M-LRSR[40]	80.0	80.2	65.5	58.3	71.0	25.3
Latent[17]+M-TJM[41]	80.1	72.9	64.5	36.2	68.4	26.9
Latent[17]+MDAN[42]	80.7	83.1	64.4	58.7	71.7	18.1
Reshape[18]+DSM[38]	86.4	91.4	68.3	60.2	76.6	19.6
Reshape[18]+DAM[39]	85.0	87.3	68.2	60.2	75.2	18.1
Reshape[18]+M-GFK[1]	71.1	72.4	58.5	52.7	63.7	21.1
Reshape[18]+M-LRSR[40]	83.8	87.4	67.3	59.6	74.5	26.2
Reshape[18]+M-TJM[41]	83.6	87.0	67	59.4	74.3	29.1
Reshape[18]+MDAN[42]	80.8	82.3	65.1	58.6	71.1	19.3
LRE-SVMs[19]	82.3	91.0	66.2	56.1	73.9	19.5
Ours+DSM[38]	**86.7**	**94.7**	**69.5**	61.4	**78.1**	**34.4**
Ours+DAM[39]	85.6	94.5	68.2	60.5	77.2	29.6
Ours+M-GFK[1]	78.3	82.1	65.5	56.5	70.6	31.9
Ours+M-LRSR[40]	84.3	89.4	68.5	60.7	75.7	30.8
Ours+M-TJM[41]	84.8	88.7	67.7	**61.7**	75.7	33.8
Ours+MDAN[42]	82.9	88.8	66.3	59.2	74.3	29.4

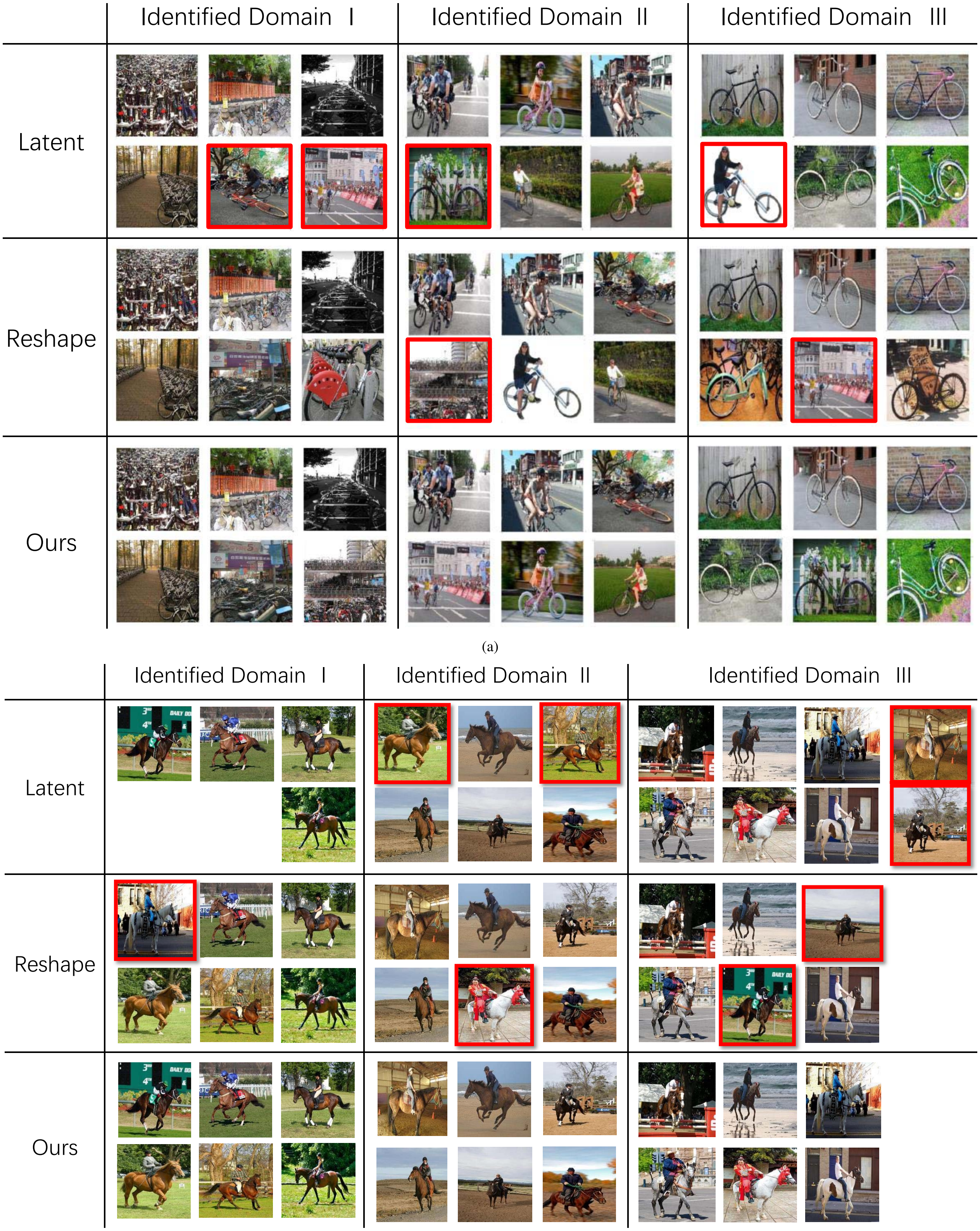
TABLE II
ACCURACIES (%) OF DIFFERENT METHODS OF DEEP DOMAIN ADAPTATION ON THE OBJECT DATASET AND THE ACTION DATASET

	Object Classification					Action Recognition
Source domain	**CPB**	**IPB**	**ICB**	**ICP**	Avg	**Stanford40**
Target domain	**I**	**C**	**P**	**B**		**UCF101**
CNN[30]	80.7	80.6	64.0	57.6	70.7	15.7
DDC[22]	79.1	83.2	60.1	57.8	70.1	19.3
DAN[23]	81.3	87.6	65.7	57.9	73.1	28.6
RevGrad[25]	82.5	89.5	64.9	58.5	76.8	25.9
RTN[26]	81.5	86.5	65.2	58.7	73.0	29.4
JAN[24]	81.6	86.8	64.4	58.4	72.8	31.1
AutoDIAL[43]	80.0	85.0	60.4	56.2	70.4	28.2
WDGRL[44]	80.3	85.2	65.2	57.8	72.1	29.2
mDA[21]	81.3	85.2	60.4	56.4	70.8	28.3
Ours+DSM[38]	**86.7**	**94.7**	**69.5**	**61.4**	**78.1**	**34.4**

(Stanford40 dataset) which makes this task more challenging.

2) Comparison With Deep Domain Adaptation Methods: Table II shows the accuracy comparison between a variety of deep domain adaptation methods and our method on the object and action datasets. From Table II, it is noticeable that the recognition accuracies of the proposed method are higher than those of other state-of-the-art methods on both object and action datasets, which clearly verifies the benefit of exploiting the multiple latent domains for domain adaptation by capturing the intrinsic structure of source data. In other words, it is beneficial to separate the source data into multiple domains to further improve the recognition accuracy on the target domain. Compared with the deep method of mDA [21], which is also able to discover latent domains for domain adaptation, our method still achieves a much better result probably due to the employment of multiple source-domain adaptation to adaptively combine multiple source classifiers for classification on the target domain.

3) Qualitative Analysis of the Discovered Latent Domains: Fig. 4(a) shows exemplar images corresponding to different discovered latent domains by different methods on the



(a)

(b)

Fig. 4. Exemplar images from the identified latent domains by different methods of Latent (top row), Reshape (middle row), and Ours method (bottom row) on both (a) object classification and (b) action recognition. (This figure is best viewed in color.)

object dataset. All of these images are of the same bike class. Note that three identified domains generally correspond to different semantic meanings. Domain I mainly corresponds to a bunch of bikes, Domain II mainly corresponds to bicycling, and Domain III mainly corresponds to a single bicycle. Compared with the methods of Latent and Reshape, Ours

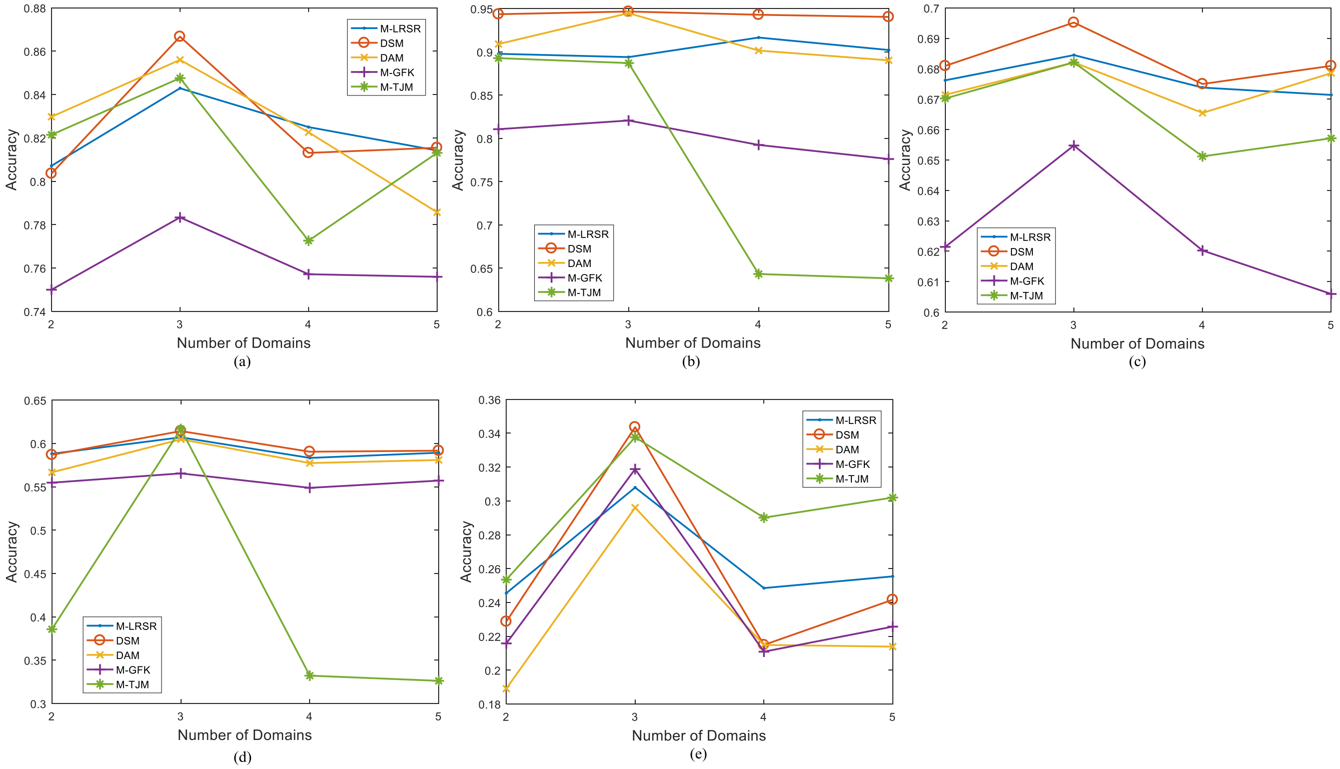


Fig. 5. Results of different domain numbers on object classification and action recognition. (a) Object classification (CPB→I). (b) Object classification (IPB→C). (c) Object classification (ICB→P). (d) Object classification (ICP→B). (e) Action recognition.

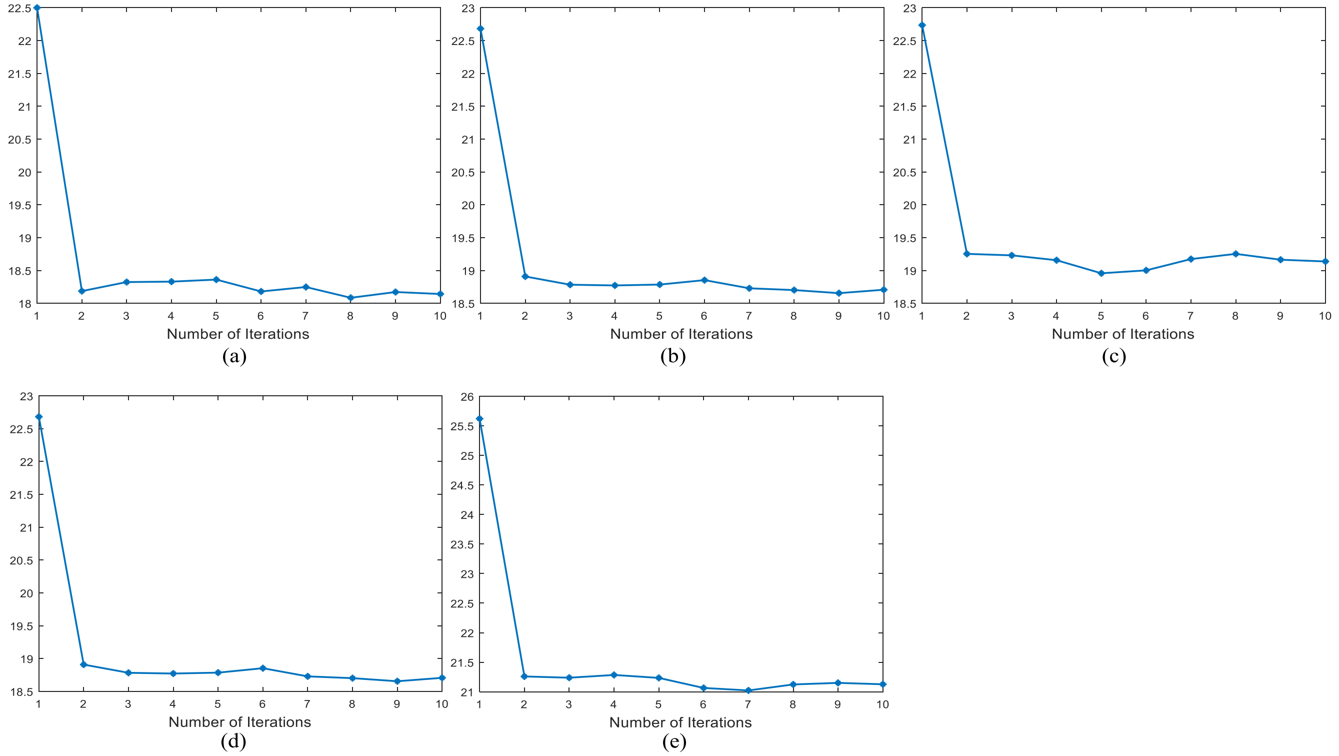


Fig. 6. Illustration of the convergence of our method. (a) Object classification (CPB→I). (b) Object classification (IPB→C). (c) Object classification (ICB→P). (d) Object classification (ICP→B). (e) Action recognition.

method performs best since all exemplar images are classified into the correct domains. For the methods of Latent (top row) and Reshape (middle row), several images (denoted by red bounding boxes) are not correctly classified into their corresponding latent domains. Fig. 4(b) shows an example of the horse-riding class in the action dataset. The identified Domains

I–III correspond to the scenes of grassy lawn, wild desert, and urban environment, respectively. It is interesting to observe that our method can automatically assign almost all instances to their correct latent domains.

4) Quantitative Evaluations on the Number of Latent Domains: Additional experiments are conducted on the object and action datasets to study how the group number affects the domain adaptation performance. Fig. 5 illustrates the recognition accuracies of our method with respect to the increasing number of latent domains. It is obvious that for both object recognition and action recognition, the result first increases and then declines with the increasing domain number. So the optimal number of latent domains is empirically set to 3.

5) Convergence of the Iterative Algorithm: Fig. 6 experimentally demonstrates the convergence of the proposed iterative algorithm to jointly learn multiple latent domains and deep representations for both object classification and action recognition. It is evident that the learned deep features converge after fewer than five iterations.

V. Conclusion

A novel deep neural-network-based approach has been presented to discover multiple latent domains from source data for domain adaptation. Different from previous methods, we simultaneously address latent domain discovery and visual feature learning in a unified deep architecture which is learned in an end-to-end fashion. A new iteration optimization algorithm is presented to learn the deep model, which alternates between applying a clustering method to predict the domain labels and training the deep neural networks using the predicted domain label as a supervision. In the training stage, a domain prediction loss, a task-specific classification loss, and a domain confusion loss are effectively combined into the objective function, which makes the learned feature domain distinguishable, semantically meaningful, and domain transferable. Extensive experiments on both object classification and action recognition demonstrate the efficacy of the proposed model against existing methods.

References

[1] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[2] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 999–1006.

[3] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016, pp. 2058–2065.

[4] 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.

[5] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[6] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017.

[7] M. Uzair and A. S. Mian, "Blind domain adaptation with augmented extreme learning machine features," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 651–660, Mar. 2017.

[8] Y. Chen, S. Song, S. Li, L. Yang, and C. Wu, "Domain space transfer extreme learning machine for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1909–1922, May 2019.

[9] C.-X. Ren, X.-L. Xu, and H. Yan, "Generalized conditional domain adaptation: A causal perspective with low-rank translators," *IEEE Trans. Cybern.*, to be published.

[10] X. Wang, W. Huang, Y. Cheng, Q. Yu, and Z. Wei, "Multisource domain attribute adaptation based on adaptive multikernel alignment learning," *IEEE Trans. Cybern.*, to be published.

[11] A. Raj, V. P. Namboodiri, and T. Tuytelaars, "Subspace alignment based domain adaptation for RCNN detector," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–11.

[12] J. Hoffman *et al.*, "LSDA: Large scale detection through adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3536–3544.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[14] W. Li, L. Chen, D. Xu, and L. Van Gool, "Visual recognition in RGB images and videos by learning from RGB-D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 2030–2036, Aug. 2018.

[15] J. Zhang, Y. Han, J. Tang, Q. Hu, and J. Jiang, "Semi-supervised image-to-video adaptation for video action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 960–973, Apr. 2017.

[16] 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. 2014.

[17] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[18] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1286–1294.

[19] 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.

[20] C. Xiong, S. McCloskey, S.-H. Hsieh, and J. J. Corso, "Latent domains modeling for visual domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2860–2866.

[21] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci, "Boosting domain adaptation by discovering latent domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3771–3780.

[22] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014. [Online]. Available: http://arxiv.org/abs/1412.3474

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

[24] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

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

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[27] 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.

[28] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[29] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8503–8512.

[30] 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.

[31] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[32] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[33] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[34] A. Bergamo and L. Torresani, "Exploiting weakly-labeled Web images to improve object classification: A domain adaptation approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 181–189.

[35] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1331–1338.

[36] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[37] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2799–2813, Dec. 2018.

[38] L. Duan, D. Xu, and S.-F. Chang, "Exploiting Web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1338–1345.

[39] L. Duan, D. Xu, and I. W.-H. 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.

[40] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, Feb. 2016.

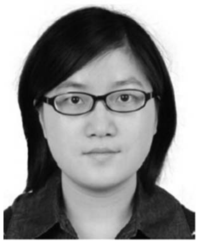
[41] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1410–1417.

[42] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8559–8570.

[43] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "AutoDIAL: Automatic domain alignment layers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5077–5085.

[44] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4058–4065.

[45] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

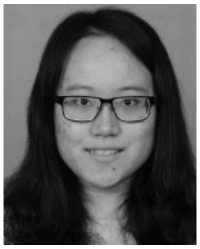


Xinxiao Wu (M'09) received the B.S. degree in computer science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005 and the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2010.

She is an Associate Professor with the School of Computer Science, Beijing Institute of Technology. She was a Postdoctoral Research Fellow with Nanyang Technological University, Singapore, from 2010 to 2011. Her current research interests include machine learning, computer vision, and video analysis and understanding.

Jin Chen received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2017, where she is currently pursuing the Ph.D. degree in computer science with the Beijing Laboratory of Intelligent Information Technology.

Her current research interests include domain adaptation, reinforcement learning, and machine learning.

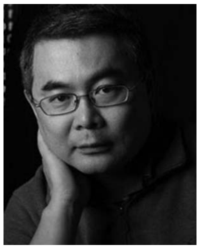


Feiwu Yu received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2016, where she is currently pursuing the M.S. degree in computer science with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science.

Her current research interests include action recognition and transfer learning.

Mingyu Yao received the B.S. degree in computer science from Zhengzhou University, Zhengzhou, China, in 2015 and the M.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2018.

Her current research interests include domain adaptation, transferring learning, and machine learning.



Jiebo Luo (S'93–M'96–SM'99–F'09) received the B.S. and M.S. degrees from the University of Science and Technology of China, Hefei, China, in 1989 and 1992, respectively, and the Ph.D. degree from the University of Rochester, Rochester, NY, USA, in 1995.

He joined the Department of Computer Science, University of Rochester, Rochester, NY, USA, in 2011, after a prolific career of over 15 years with Kodak Research Labs, Rochester, NY, USA. He has authored over 400 technical papers and holds over 90 U.S. patents. His current research interests include computer vision, machine learning, data mining, social media, and biomedical informatics.

Mr. Luo has served as the Program Chair of the ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, and on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON BIG DATA, *Pattern Recognition*, *Machine Vision and Applications*, and *ACM Transactions on Intelligent Systems and Technology*. He is also a fellow of ACM, AAAI, SPIE, and IAPR.