IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2020, accepted December 22, 2020, date of publication December 25, 2020,
date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047448

Extreme Learning Machine Based on Maximum Weighted Mean Discrepancy for Unsupervised Domain Adaptation

YANNA SI[1], **JIEXIN PU**[1], **SHAOFEI ZANG**[1], **AND LIFAN SUN**[1,2]
[1]School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China
[2]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Lifan Sun (lifan.sun@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U1504619, in part by the Aeronautical Science Foundation of China under Grant 20185142003, and in part by the National Defense Basic Scientific Research Program of China under Grant JCKY2018419C001.

ABSTRACT Extreme Learning Machine (ELM) has shown fast learning speed and good generalization property in single-domain problems, such as classification and regression. However, the assumption that the training and testing data are subject to identical distribution often leads to significant performance degradation of ELM in cross-domain problems. To cope with unsupervised domain adaptation problems by ELM, we propose a novel method called Extreme Learning Machine based on Maximum Weighted Mean Discrepancy (ELM-MWMD) in this paper, which learns an adaptive ELM classifier with both labeled source data and unlabeled target data. Firstly, the cross-domain weight coefficients are specifically designed and assigned for each sample in source and target domains, fully considering the effects of individual information. Then the source classifier is adapted to the target domain by minimizing the distribution discrepancy between the two domains, both the marginal distribution and conditional distribution are simultaneously reduced to obtain a more accurate target classifier. Moreover, the predicted results for target data are utilized as pseudo labels to further improve the classification accuracy in multiple iterations. Extensive experiments on public image datasets demonstrate that ELM-MWMD performs better than several existing state-of-the-art domain adaptation methods by computation efficiency and classification accuracy.

INDEX TERMS Extreme learning machine, domain adaptation, cross-domain weight, maximum mean discrepancy, joint distribution adaptation.

I. INTRODUCTION

In the current era of big data, a huge number of image, text, audio and video data can be obtained from different fields. However, only a small fraction of them that are annotated can play a part, while most original data is discarded for lack of accurate label information. Annotating data with high quality is laborious and expensive, which has not been completely solved yet. On the other hand, it is usual in most of the machine learning tasks to assume that the training and testing data follow identical distribution, which is often violated in practical applications. Transfer learning has provided a good solution to these problems and achieved fruitful results, such as in computer vision [1]– [3], medical image analysis [4], [5] and natural language processing [6], [7], etc. Domain adaptation is a representative method in transfer learning, which can leverage the labeled source data to develop a model for the different but related target domain, in the case that the target domain has little or even no labels.

Neural networks have been widely used in main adaptation research for the superior performance [8]–[10]. As a special single layer feedforward neural network (SLFN) proposed by Huang *et al.* [11], Extreme Learning Machine (ELM) has faster learning speed and better generalization than Support Vector Machine (SVM) and back propagation (BP) neural network for the regression and classification tasks [12]. In some practical applications, e.g., face recognition [13], [14], ELM also shows excellent performance and efficiency. Although classical ELM handles machine learning tasks well in single domain, it cannot

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao.

directly deal with the cross-domain problems, where the training and testing data are from different distributions.

In order to take full advantage of ELM, some modified methods have been incorporated into domain adaptation research over the years. Zhang and Zhang [15] are the first to study the application of ELM in domain adaptation problems. They have put forward the domain adaptation extreme learning machine (DAELM) framework, and on this basis, two different methods have been proposed to learn a robust classifier for gas identification and drift compensation of e-nose system. Subsequently, the work in [16] has combined the theory of online sequential extreme learning machine with DAELM to establish the online drift compensation model, so that the recognition models could adapt to the changes of sensor responses in a time-efficient manner without losing the high accuracy. Besides, a parameter transfer-based domain adaption ELM has been developed in [17], which projected the target parameters to the source and made the parameters maximally aligned, thus the parameters of the classifier and the transformation matrix could be calculated simultaneously.

However, the above mentioned methods require some labeled data in the target domain, which is consistent with the restriction of semi-supervised learning. To deal with the cases where there is no labeled data in target domain, Liu *et al.* [18] have proposed a unified subspace transfer framework called cross-domain extreme learning machine (CdELM), in which the ELM was well-exploited to learn a shared subspace across domains. Furthermore, Chen *et al.* [19] have developed another ELM-based space learning algorithm, domain space transfer ELM (DST-ELM), both source and target data were reconstructed in a domain invariant space, and the distribution distance of the two domains was also minimized here. Instead of learning a shared subspace by ELM, Zhang *et al.* [20] have integrated the scalable factor into discriminative ELM (DELM) to strengthen the discriminative capacity of the ELM classifier, and proposed a joint unsupervised cross-domain model via scalable discriminative ELM (JUC-SDELM). In addition, Li *et al.* [21] have proposed a cross-domain ELM framework for unsupervised domain adaptation, in which the source classifier was adapted to target domain by matching the projected means of both domains, and the structural property of the target domain was explored by manifold regularization to make the final classifier more adaptable.

In the mentioned feature-based domain adaptation methods, a nonparametric estimate criterion of distance, Maximum Mean Discrepancy (MMD) has been frequently used to reduce the distribution difference between domains. It measures the probability distribution distance by the maximum mean function rather than solving an intermediate density estimation procedure, which is simpler and more applicable for domain adaptation problems [22], [23]. Nevertheless, the existing MMD-based domain adaptation methods just show the whole distribution information and global structural information of the data space, the sample individual information is neglected.

Motivated by the sample weight mechanism presented in [24], [25], we put forward a novel method, ELM-MWMD, to deal with the unsupervised domain adaptation problems. In our method, the individual difference of samples is fully considered and utilized to learn a more adaptive target classifier. The basic idea of ELM-MWMD is illustrated in Fig 1. Firstly, we assign the cross-domain weight to each sample in both domains, which can effectively reflect the contribution difference of samples. Then the MWMD is utilized to estimate and minimize the distribution discrepancy between the source and target domains, thus the source classifier learned with the labeled data can adapt to target domain well. In MWMD, the marginal distribution and conditional distribution differences between domains are simultaneously considered, and the predicted results are iteratively applied to the ELM classifier to further improve the classification accuracy.

The contributions of this paper are summarized as follows:

(1) The different effects of individual sample are explicitly considered and we design special cross-domain weights for each sample, which is beneficial in reducing calculation burden.

(2) Combining the cross-domain weights with MMD, the MWMD is proposed to measure and minimize the distribution difference of source and target domain.

(3) We incorporate ELM into the MWMD, and propose an ELM-MWMD to learn a cross-domain classifier with labeled source data and unlabeled target data in unsupervised domain adaptation problems.

(4) Comprehensive experiments are conducted on real-world image datasets, including Office, Caltech-256, MNIST, USPS and COIL20 to verify the efficiency of the proposed ELM-MWMD.

The remainder of the paper is organized as follows. In section 2, we briefly reviewed the basics of ELM and DA. Then the ELM-MWMD with a detailed model and optimization algorithm is presented in Section 3. Section 4 describes the contrast experiments and the results analysis. The conclusions are finally made in section 5.

II. PRELIMINARIES

In this section, the basic knowledge of the Extreme Learning Machine (ELM) and Domain Adaptation (DA) is briefly introduced.

A. EXTREME LEARNING MACHINE (ELM)

Different from the conventional feedforward neural networks, ELM is a special single-hidden layer feedforward neural network (SLFN), which randomly initializes the input weight and bias of hidden nodes without tuning. The output weights of ELM can be analytically solved in a simple way rather than computing the gradient iteratively.

Given a training set $\left\{\boldsymbol{x}_i, \boldsymbol{y}_i\right\}_{i=1}^{N}$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$ and $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \cdots, y_{im}]^T \in R^m$. The output of the standard SLFNs with L hidden nodes and

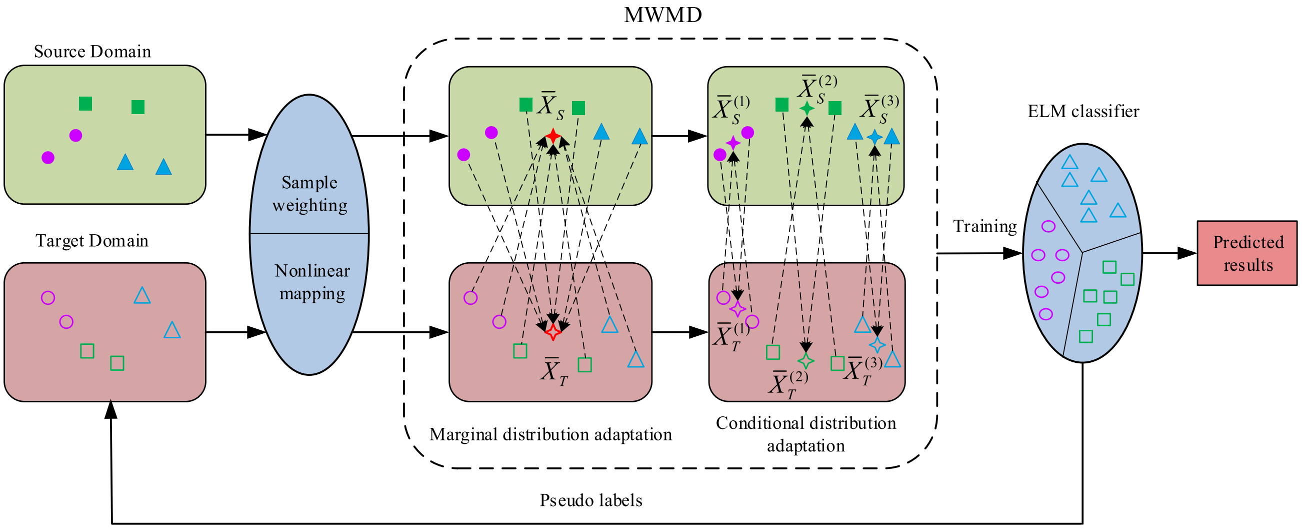


FIGURE 1. Illustration of the ELM-MWMD method. $\bar{X}$ is the mean point of samples and $\bar{X}^{(c)}$ is the mean point of samples belong to class in source or target domain.

activation function $h(x)$ is defined as:

$$\boldsymbol{o}_j = \sum_{i=1}^{L} \beta_i h\left(\boldsymbol{w}_i \cdot \boldsymbol{x}_j + b_i\right), \quad j = 1, 2, \cdots, N \tag{1}$$

where $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ is the input weight vector, $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the output weight vector and b_i is the bias term of the ith hidden node.

In a supervised learning problem, the SLFNs can approximate the samples with zero error, namely the output is equal to the expected value,

$$\boldsymbol{y}_j = \boldsymbol{o}_j = \sum_{i=1}^{L} \beta_i h\left(\boldsymbol{w}_i \cdot \boldsymbol{x}_j + b_i\right) \tag{2}$$

For the sake of simplicity, there is a concise matrix format of the network outputs:

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{Y} \tag{3}$$

where $\boldsymbol{H}_{N\times L}$ is the output matrix of the hidden layer, $\boldsymbol{\beta}_{L\times m}$ is the output weight matrix of the network, and

$$\boldsymbol{H} = \begin{bmatrix} h(w_1 \cdot x_1 + b_1) & \cdots & h(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ h(w_1 \cdot x_N + b_1) & \cdots & h(w_L \cdot x_N + b_L) \end{bmatrix},$$

$$\boldsymbol{\beta} = \left[\beta_1^T, \beta_2^T, \cdots, \beta_L^T\right]^T, \boldsymbol{Y} = \left[y_1^T, y_2^T, \cdots, y_N^T\right]^T \tag{4}$$

Generally, $\boldsymbol{\beta}$ can be solved by minimizing the sum of squared losses of prediction errors. In ELM algorithm, it is expressed as the following regularized least square optimization problem:

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|\xi_i\|^2$$
$$s.t.\ \boldsymbol{h}(x_i)\,\boldsymbol{\beta} = \boldsymbol{y}_i^T - \xi_i^T, \quad i = 1, 2, \cdots, N \tag{5}$$

where C is a relevant penalty factor, $\xi_i \in R^m$ is the prediction error vector of the ith sample.

Further, the problem (5) can be converted into an unconstrained optimization problem as

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\beta}\|^2 \tag{6}$$

This problem can be determined analytically by the Moore-Penrose generalized inverse and has a closed form solution as follows:

$$\boldsymbol{\beta}^* = \begin{cases} \boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T + \dfrac{\boldsymbol{I}_N}{C}\right)^{-1}\boldsymbol{Y}, & N < L \\ \left(\boldsymbol{H}^T\boldsymbol{H} + \dfrac{\boldsymbol{I}_L}{C}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}, & N \geq L \end{cases} \tag{7}$$

where $\boldsymbol{I}_N$ and $\boldsymbol{I}_L$ represent the identity matrix of the corresponding dimensions.

It can be seen that the parameters of hidden layer in ELM are not updated iteratively, and as a result of (7), the output weight can be easily solved without complicated back propagation, which makes the ELM perform more efficient than other BP networks.

B. DOMAIN ADAPTATION

Domain adaptation is developed to deal with the scenarios that the training data and testing data come from different domains. According to the different types of target and source domain, there are four different types of domain adaptation problems: unsupervised, supervised, heterogeneous distribution and multiple source domains [26].

In this paper, we aim at addressing the unsupervised domain adaptation problems, in which the training data is sampled from the source domain with accurate label

TABLE 1. Notations in domain adaptation problems.

Terminology	Source	Target
Domain	$\mathcal{D}_S = \{\mathcal{X}_S, P(\boldsymbol{X}_S)\}$	$\mathcal{D}_T = \{\mathcal{X}_T, P(\boldsymbol{X}_T)\}$
Data	$\boldsymbol{X}_S = \{(\boldsymbol{x}_{Si}, \boldsymbol{y}_i)\}_{i=1}^{n_S}$	$\boldsymbol{X}_T = \{(\boldsymbol{x}_{Tj})\}_{j=1}^{n_T}$
Feature space	$\mathcal{X}_S \in R^{n_S \times d}$	$\mathcal{X}_T \in R^{n_T \times d}$
Label space	$\mathcal{Y}_S \in R^{n_S \times m}$	$\mathcal{Y}_T \in R^{n_T \times m}$
Marginal distribution	$P(\boldsymbol{X}_S)$	$P(\boldsymbol{X}_T)$
Conditional distribution	$P(\boldsymbol{Y}_S \mid \boldsymbol{X}_S)$	$P(\boldsymbol{Y}_T \mid \boldsymbol{X}_T)$

information and the testing data sampled from the target domain is fully unlabeled.

We define the source domain data $\boldsymbol{X}_S = \left\{\left(\boldsymbol{x}_{Si}, \boldsymbol{y}_i\right)\right\}_{i=1}^{n_S}$ and the target domain data $\boldsymbol{X}_T = \left\{\left(\boldsymbol{x}_{Tj}\right)\right\}_{j=1}^{n_T}$, where n_S and n_T are the sample numbers, respectively. The source data and the target data belong to the same feature space $\mathcal{X}_S = \mathcal{X}_T$ and label space $\mathcal{Y}_S = \mathcal{Y}_T$, the marginal distribution and conditional distribution are different. Table 1 is a list of some primary notations in domain adaptation problems.

Maximum Mean Discrepancy (MMD) is a nonparametric estimate criterion of distance. Compared with the Bregman divergence [27] and the Kullback-Leibler (K-L) divergence [28], it measures the difference between two probability distributions via the maximum mean function rather than solving an intermediate density estimation procedure [23]. So it is more simple and applicable for minimizing the distribution distance between the source and target domains. Generally, the empirical estimate of MMD in a Reproducing Kernel Hilbert Space (RKHS) is utilized for DA problems.

$$MMD^2\left[\boldsymbol{X}_S, \boldsymbol{X}_T\right] = \left\| \frac{1}{n_S}\sum_{i=1}^{n_S} \phi\left(x_{Si}\right) - \frac{1}{n_T}\sum_{j=1}^{n_T} \phi\left(x_{Tj}\right) \right\|_{\mathcal{H}}^2 \tag{8}$$

where $\phi(\cdot)$ is the mapping kernel function, $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm.

III. ELM-MWMD

In this section, the proposed ELM-MWMD method is described in details. It is developed to deal with the unsupervised domain adaptation problems, where the distribution of source data and target data are related and they share the same category information. Specially, the cross-domain weight strategy is introduced to reflect the effects of individual sample in causing the distribution differences and the discrepancy of the marginal and conditional distributions are simultaneously minimized to obtain a more adaptable target classifier.

A. ELM SOURCE CLASSIFIER

In domain adaptation problems, there are accurate labels only in source domain, where the classifier can be easily learned by supervised learning. So we firstly adopt ELM to learn the source classifier, due to its better learning efficiency and generalization performance than other ordinary algorithms. The regularized formulation of ELM is expressed as following:

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{n_S}\|\boldsymbol{e}_{Si}\|^2$$
$$s.t.\ \boldsymbol{h}\left(\boldsymbol{x}_{Si}\right)\boldsymbol{\beta} = \boldsymbol{y}_{Si}^T - \boldsymbol{e}_{Si}^T \tag{9}$$

By substituting the constraints into the objective function, an equivalent unconstrained optimization problem can be obtained as:

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{n_S}\|\boldsymbol{Y}_S - \boldsymbol{H}_S\boldsymbol{\beta}\|^2 \tag{10}$$

where $\boldsymbol{H}_S = \left[\boldsymbol{h}\left(x_{S1}\right); \boldsymbol{h}\left(x_{S2}\right); \cdots; \boldsymbol{h}\left(x_{Sn_S}\right)\right] \in R^{n_S \times L}$ is the hidden layer output matrix and $\boldsymbol{Y}_S = \left[\boldsymbol{y}_{S1}^T, \boldsymbol{y}_{S2}^T, \cdots, \boldsymbol{y}_{Sn_S}^T\right]^T \in R^{n_S \times m}$ is the label matrix.

We rewrite it in an equivalent form

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}Tr\left(\left(\boldsymbol{H}_S\boldsymbol{\beta} - \boldsymbol{Y}_S\right)^T\left(\boldsymbol{H}_S\boldsymbol{\beta} - \boldsymbol{Y}_S\right)\right) \tag{11}$$

where $Tr(\ \cdot\ )$ is the trace of a matrix.

We hope that the classifier could work with high-efficiency not only in source domain, but also in target domain. However, it cannot be applied directly to the target domain, because of the distribution difference between domains. Reducing the discrepancy of the two domains is a feasible approach to generalize the source classifier to the target data [29], which has been widely applied in domain adaptation research.

B. DISTRIBUTION ADAPTATION

MMD is a nonparametric metric, which has been frequently used in domain adaptation problems. There, the projected MMD is employed to measure and decrease the distribution distance. In addition, the marginal distribution and conditional distribution are simultaneously considered according to the Joint Distribution Adaptation (JDA) strategy presented in [30].

Firstly, we compute the distance between the marginal distributions:

$$D^2\left[\boldsymbol{X}_S, \boldsymbol{X}_T\right] = \left\| \frac{1}{n_S}\sum_{i=1}^{n_S} \boldsymbol{h}\left(\boldsymbol{x}_i\right)\boldsymbol{\beta} - \frac{1}{n_T}\sum_{j=1}^{n_T} \boldsymbol{h}\left(\boldsymbol{x}_j\right)\boldsymbol{\beta} \right\|_{\mathcal{H}}^2$$
$$= Tr\left(\boldsymbol{\beta}^T\boldsymbol{H}^T\boldsymbol{M}_0\boldsymbol{H}\boldsymbol{\beta}\right) \tag{12}$$

where $\boldsymbol{H} = \left[\boldsymbol{H}_S^T, \boldsymbol{H}_T^T\right]^T \in R^{(n_S+n_T)\times L}$ is the nonlinear mapping matrix for all the source and target samples, $\boldsymbol{M}_0$ is the MMD matrix which can be computed by

$$\left(\boldsymbol{M}_0\right)_{ij} = \begin{cases} \dfrac{1}{n_S n_S}, & \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_S \\ \dfrac{1}{n_T n_T}, & \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_T \\ \dfrac{-1}{n_S n_T}, & otherwise \end{cases} \tag{13}$$

Then the modified MMD is used to measure the distance between the class-conditional distributions:

$$\begin{aligned} &D^2[\boldsymbol{X}_S, \boldsymbol{X}_T]_c \\ &= \sum_{c=1}^{C} \left\| \frac{1}{n_S^{(c)}} \sum_{x_i \in \mathcal{D}_S^{(c)}} \boldsymbol{h}\left(x_i^{(c)}\right)\boldsymbol{\beta} - \frac{1}{n_T^{(c)}} \sum_{x_j \in \mathcal{D}_T^{(c)}} \boldsymbol{h}\left(x_j^{(c)}\right)\boldsymbol{\beta} \right\|_{\mathcal{H}}^2 \\ &= Tr\left(\boldsymbol{\beta}^T \boldsymbol{H}^T \left(\sum_{c=1}^{C} \boldsymbol{M}_c\right) \boldsymbol{H\beta}\right) \end{aligned} \tag{14}$$

where $\boldsymbol{D}_S^{(c)} = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in \mathcal{D}_S \cap \mathcal{Y}(\boldsymbol{x}_i) = c\}$ is the set of source examples in class c and $\mathcal{Y}(\boldsymbol{x}_i)$ is the true label of $\boldsymbol{x}_i$; correspondingly,$\boldsymbol{D}_T^{(c)} = \left\{\boldsymbol{x}_j : \boldsymbol{x}_j \in \mathcal{D}_T \cap \mathcal{Y}\left(\boldsymbol{x}_j\right) = c\right\}$ is the predicted label of $\boldsymbol{x}_j$.

And the MMD matrix within classes can be computed by

$$(\boldsymbol{M}_c)_{ij} = \begin{cases} \dfrac{1}{n_S^{(c)} n_S^{(c)}}, & \boldsymbol{x}_i^{(c)}, \boldsymbol{x}_j^{(c)} \in \mathcal{D}_S^{(c)} \\ \dfrac{1}{n_T^{(c)} n_T^{(c)}}, & \boldsymbol{x}_i^{(c)}, \boldsymbol{x}_j^{(c)} \in \mathcal{D}_T^{(c)} \\ \dfrac{-1}{n_S^{(c)} n_T^{(c)}}, & \begin{cases} \boldsymbol{x}_i^{(c)} \in \mathcal{D}_S^{(c)} and \boldsymbol{x}_j^{(c)} \in \mathcal{D}_T^{(c)} \\ \boldsymbol{x}_j^{(c)} \in \mathcal{D}_S^{(c)} and \boldsymbol{x}_i^{(c)} \in \mathcal{D}_T^{(c)} \end{cases} \\ 0, & otherwise \end{cases} \tag{15}$$

Based on the above analysis, we incorporate (12) and (14) to get the joint distribution adaptation optimization function:

$$\begin{aligned} &\min_{\beta} Tr\left(\left(\boldsymbol{\beta}^T \boldsymbol{H}^T \boldsymbol{M}_0 \boldsymbol{H\beta}\right) + \left(\boldsymbol{\beta}^T \boldsymbol{H}^T \left(\sum_{c=1}^{C} \boldsymbol{M}_c\right) \boldsymbol{H\beta}\right)\right) \\ &\qquad = \min_{\beta} Tr\left(\boldsymbol{\beta}^T \boldsymbol{H}^T \left(\boldsymbol{M}_0 + \sum_{c=1}^{C} \boldsymbol{M}_c\right) \boldsymbol{H\beta}\right) \end{aligned} \tag{16}$$

Minimizing the distance between the marginal distributions may ignore the class information across domains, so we have also considered the conditional distribution difference. It can transfer the well-labeled source class knowledge to the target domain, then improve the final target classification.

C. CROSS-DOMAIN WEIGHTS

Based on the above analysis, MMD reveals the distribution difference by solving the mean error of the two datasets, only considering the whole data distribution and ignoring the individual sample information. For the purpose of reflecting the effects of individual sample in causing distribution differences, we introduce a cross-domain weight for each sample. Combining the cross-domain weights with MMD, the MWMD is proposed to further measure and minimize the distribution difference between domains.

For the marginal distribution, the optimized objective function can be expressed as

$$\begin{aligned} J_1(\boldsymbol{\beta}) &= \min_{\beta} \left\| \begin{array}{l} \dfrac{1}{n_S} \displaystyle\sum_{i=1}^{n_S} \left(\boldsymbol{h}(\boldsymbol{x}_i) + w_i \boldsymbol{h}(\boldsymbol{x}_i)\right)\boldsymbol{\beta} \\ -\dfrac{1}{n_T} \displaystyle\sum_{j=1}^{n_T} \left(\boldsymbol{h}(\boldsymbol{x}_j) + w_j \boldsymbol{h}(\boldsymbol{x}_j)\right)\boldsymbol{\beta} \end{array} \right\|_{\mathcal{H}}^2 \\ &= \min_{\beta} Tr\left(\boldsymbol{\beta}^T \boldsymbol{H}^T (\boldsymbol{M}_0 + \boldsymbol{W}_0) \boldsymbol{H\beta}\right) \end{aligned} \tag{17}$$

where the weight matrices $\boldsymbol{W}_0$ is defined as follows:

$$(\boldsymbol{W}_0)_{ij} = \begin{cases} \dfrac{w_i w_j}{n_S n_S}, & \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_S \\ \dfrac{w_i w_j}{n_T n_T}, & \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_T \\ \dfrac{-w_i w_j}{n_S n_T}, & otherwise \end{cases} \tag{18}$$

where $w_i = \sqrt{\sum_{i=1}^{n_S+n_T} \left(\boldsymbol{x}_i - \bar{X}\right)^2}$ is the weight coefficient of ith sample and $\bar{X} = \begin{cases} \bar{X}_T, \boldsymbol{x}_i \in \mathcal{D}_S \\ \bar{X}_S, \boldsymbol{x}_i \in \mathcal{D}_T \end{cases}$ is the mean point of samples in source or target domain.

For the conditional distribution, the optimized objective function can be expressed as

$$\begin{aligned} &J_2(\boldsymbol{\beta}) \\ &= \min_{\beta} \sum_{c=1}^{C} \left\| \begin{array}{l} \dfrac{1}{n_S^{(c)}} \displaystyle\sum_{\boldsymbol{x}_i \in D_S^{(c)}} \left(\boldsymbol{h}\left(\boldsymbol{x}_i^{(c)}\right)\boldsymbol{\beta} + w_i^{(c)} \boldsymbol{h}\left(\boldsymbol{x}_i^{(c)}\right)\boldsymbol{\beta}\right) \\ -\dfrac{1}{n_T^{(c)}} \displaystyle\sum_{\boldsymbol{x}_j \in D_T^{(c)}} \left(\boldsymbol{h}\left(\boldsymbol{x}_j^{(c)}\right)\boldsymbol{\beta} + w_j^{(c)} \boldsymbol{h}\left(\boldsymbol{x}_j^{(c)}\right)\boldsymbol{\beta}\right) \end{array} \right\|_{\mathcal{H}}^2 \\ &= \min_{\beta} Tr\left(\boldsymbol{\beta}^T \boldsymbol{H}^T \left(\sum_{c=1}^{C} (\boldsymbol{M}_c + \boldsymbol{W}_c)\right) \boldsymbol{H\beta}\right) \end{aligned} \tag{19}$$

where the class weight matrix $\boldsymbol{W}_c$ is defined as follows:

$$(\boldsymbol{W}_c)_{ij} = \begin{cases} \dfrac{w_i^{(c)} w_j^{(c)}}{n_S^{(c)} n_S^{(c)}}, & \boldsymbol{x}_i^{(c)}, \boldsymbol{x}_j^{(c)} \in \mathcal{D}_S^{(c)} \\ \dfrac{w_i^{(c)} w_j^{(c)}}{n_T^{(c)} n_T^{(c)}}, & \boldsymbol{x}_i^{(c)}, \boldsymbol{x}_j^{(c)} \in \mathcal{D}_T^{(c)} \\ \dfrac{-w_i^{(c)} w_j^{(c)}}{n_S^{(c)} n_T^{(c)}}, & \begin{cases} \boldsymbol{x}_i^{(c)} \in \mathcal{D}_S^{(c)} and \boldsymbol{x}_j^{(c)} \in \mathcal{D}_T^{(c)} \\ \boldsymbol{x}_j^{(c)} \in \mathcal{D}_S^{(c)} and \boldsymbol{x}_i^{(c)} \in \mathcal{D}_T^{(c)} \end{cases} \\ 0, & otherwise \end{cases} \tag{20}$$

where $w_i^{(c)} = \sqrt{\sum_{i=1}^{n_S+n_T} \left(\boldsymbol{x}_i^c - \bar{X}^{(c)}\right)^2}$ is the cross-domain weight coefficient of ith sample belongs to class c and $\bar{X}^{(c)} = \begin{cases} \bar{X}_T^{(c)}, \boldsymbol{x}_i^c \in \mathcal{D}_S^{(c)} \\ \bar{X}_S^{(c)}, \boldsymbol{x}_i^c \in \mathcal{D}_T^{(c)} \end{cases}$ is the mean point of samples belong to class c in source or target domain.

By incorporating (17) and (19), the joint MWMD optimization problem can be presented as follows:

$$J(\boldsymbol{\beta}) = J_1(\boldsymbol{\beta}) + J_2(\boldsymbol{\beta}) = \min_{\beta} Tr\left(\boldsymbol{\beta}^T \boldsymbol{H}^T \left((\boldsymbol{M}_0+\boldsymbol{W}_0)+\sum_{c=1}^{C}(\boldsymbol{M}_c+\boldsymbol{W}_c)\right)\boldsymbol{H}\boldsymbol{\beta}\right) \quad (21)$$

D. OPTIMIZATION FUNCTION

We incorporate ELM into the MWMD to obtain the final ELM-MWMD optimization function, which can be rewritten as a general form:

$$\min_{\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2} Tr\left((\boldsymbol{H\beta}-\boldsymbol{Y})^T(\boldsymbol{H\beta}-\boldsymbol{Y})\right) + \frac{\lambda}{2} Tr\left(\boldsymbol{\beta}^T\boldsymbol{H}^T\left((\boldsymbol{M}_0+\boldsymbol{W}_0)+\sum_{c=1}^{C}(\boldsymbol{M}_c+\boldsymbol{W}_c)\right)\boldsymbol{H\beta}\right) \quad (22)$$

By setting the gradient with respect to β as zero, we can get the closed solution similar to the classical ELM algorithm.

When $(n_S + n_T) < L$, it is

$$\boldsymbol{\beta}^* = \boldsymbol{H}^T\left(\boldsymbol{HH}^T + \lambda\begin{pmatrix}(\boldsymbol{M}_0+\boldsymbol{W}_0)+\\ \sum\limits_{c=1}^{C}(\boldsymbol{M}_c+\boldsymbol{W}_c)\end{pmatrix}\boldsymbol{HH}^T + \frac{I_N}{C}\right)^{-1}\boldsymbol{Y}_S^T \quad (23)$$

When $(n_S + n_T) \geq L$, it is

$$\boldsymbol{\beta}^* = \left(\boldsymbol{H}^T\boldsymbol{H} + \lambda\boldsymbol{H}^T\begin{pmatrix}(\boldsymbol{M}_0+\boldsymbol{W}_0)+\\ \sum\limits_{c=1}^{C}(\boldsymbol{M}_c+\boldsymbol{W}_c)\end{pmatrix}\boldsymbol{H} + \frac{I_L}{C}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}_S^T \quad (24)$$

According to the above analysis, the ELM-MWMD method can be summarized in Algorithm 1. It is worth noting that multiple iterations are used to eliminate the occasional negative effects and further improve the classification accuracy. Simultaneously, the predicted results are taken as the pseudo labels of target data, which can iteratively become more accurate until it converges.

IV. EXPERIMENTS

In this section, we have implemented a series of experiments on public image datasets to evaluate the proposed ELM-MWMD method. A brief description of the datasets is reported in Table 3 and some image samples are displayed in Fig 2.

A. DATASETS DESCRIPTION

Office+Caltech-256 are widely used visual benchmark datasets in domain adaptation. Office [31] dataset consists of 4652 images belonging to 31 object classes. These images are collected from three different domains: Amazon (images downloaded from online chants www.amazon.com); DSLR (high-resolution images by a digital SLR camera in realistic environments); and Webcam (low-resolutions images by a simple webcam). Caltech-256 [32] is also a standard object recognition dataset that contains 30607 images of 256 classes.

TABLE 2. The procedure of ELM-MWMD method.

```
Algorithm 1 ELM-MWMD
Input: Source samples $X_S^{n_S}$ , Source labels $\mathcal{Y}_S$ ; Target samples $X_T^{n_T}$ .
1.Set the parameters $L$, $C$, $\lambda$ and the iteration number $T$;
2.Construct the MMD matrix $\boldsymbol{M}_0$ and the weight matrix $\boldsymbol{W}_0$ , set the conditional matrixes $\boldsymbol{M}_c = 0$ and $\boldsymbol{W}_c = 0$;
3.for $t = 1 : T$ do
4.Initialize the ELM network, calculate the matrix $\boldsymbol{H}$ by (4), and compute the output weight $\boldsymbol{\beta}$ by (7);
5.Train a classifier $F$ and predict the unlabeled target data to obtain their pseudo labels $\hat{\mathcal{Y}}_T$ ;
6.Update the $\boldsymbol{M}_c$ by (15) and $\boldsymbol{W}_c$ by (19);
7.Compute the optimal weights $\boldsymbol{\beta}^*$ by (23) and (24);
8.end for
Output: The predicted output $\hat{\mathcal{Y}}_T$ .
```

TABLE 3. Description of the datasets.

Datasets	Types	Samples	Classes	Features
Amazon(A)	object	958	10	800
Caltech(C)	object	1123	10	800
DSLR(D)	object	157	10	800
Webcam(W)	object	295	10	800
MNIST	digit	2000	10	256
USPS(C)	digit	1800	10	256
COIL1	object	720	20	1024
COIL2	object	720	20	1024

As shown in Table 3, we use a reduced version of Office and Caltech datasets preprocessed in Geodesic Flow Kernel (GFK) [33]. For convenience, the four domains Amazon, Caltech, DSLR and Webcam can be replaced with letters A, C, D and W, respectively. Then, we construct 12 cross-domain tasks for evaluation, e.g., A→ C, A→D . . . W→C, and W→D.

MNIST+USPS are classical hand-written digit datasets and both include ten same classes. MNIST contains 60000 training images and 10000 testing images of size 28 × 28, and USPS has a training set of 7291 images and 2007 testing images.

In the experiments, we randomly choose 1800 images from USPS and 2000 images from MNIST and all the images are resized to 16 × 16 like in [33]. We also construct two cross-domain tasks as USPS→MNIST and MNIST→USPS.

COIL20 is another visual recognition database that consists of 20 objects. The images are taken from different angles

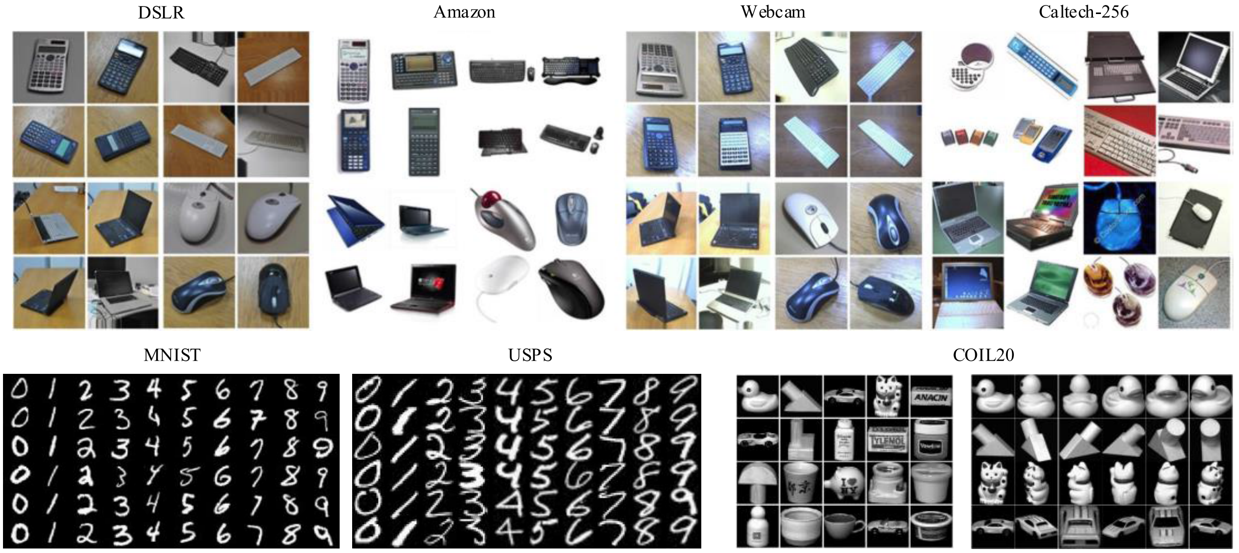


FIGURE 2. Image samples from the different datasets.

for every object. In the experiments, we utilize the demo derived from [30], in which the whole dataset is separated into two subsets COIL1 and COIL2 to build the cross-domain tasks. Both the subsets contain 720 pictures of size 32 × 32 with 20 classes. Similarly, we construct two cross-domain tasks COIL1→COIL2, COIL2→COIL1.

B. EXPERIMENTAL SETTINGS

To validate the efficiency of the proposed ELM-MWMD method, we have tested it on 16 cross-domain tasks. The results are compared with several state-of-the-art related methods, including supervised classification methods, semi-supervised and unsupervised domain adaptation methods. For Office datasets and hand-written digit datasets, two traditional classification algorithms and 5 domain adaptation approaches were compared, including SVM [34], ELM [12], TCA [35], JDA [30], DAELM (DAELM-S and DAELM-T) [15], CDELM-M [21] and DST-ELM [19]. For COIL20 dataset, we chose some other DA algorithms, such as PCA, GFK [33], SA [36] and JUC-SDELM [20]. For fairness, the experiments in every cross-domain task have been run 20 times and recorded the average value.

It should be noted that there are no labels in target domain, the parameters cannot be tuned by cross validation. So we empirically search in parameter space to get the respective optimal parameters of all methods. For classical algorithms as SVM, ELM, PCA, TCA, SA and GFK, we follow the conventional settings. For DAELM, the number of labeled target sample is set to 10. In ELM-MWMD, there are three tunable parameters: L, C and λ. The detailed parameter effects will be discussed later.

C. RESULTS AND ANALYSIS

First, we test the ELM-MWMD on Office+Caltech-256 datasets and MNIST+USPS datasets, the comparison results are displayed in Table 4, where the best results of each task are in bold. It can be seen that the average classification accuracy of ELM-MWMD is 52.80%, which is the highest in comparison to other methods. It has an improvement of 3.65% against the best baseline CDELM-M. Specifically, the ELM-MWMD performs particularly well in task MNIST→USPS and task USPS→MNIST, the average accuracy increases by 16.87% and 11.29% than the relative best baseline, respectively.

Second, we test the ELM-MWMD on COIL20 dataset with some other baselines that are different from Office dataset. Table 5 summarizes the accuracy of all the methods and the best results are still in bold. It is obvious that the ELM-MWMD outperforms the other baselines in both cross-domain tasks. The average accuracy is 2.08% higher than the best baseline, JUC-SDELM. Though the superiority is not great, it still verifies that the ELM-MWMD can achieve domain adaptation well.

For better interpretation, Fig 3 gives the classification accuracy results of all cross-domain tasks on three datasets, and it can be seen that: 1) The two supervised learning algorithms, SVM and ELM, fail to achieve satisfactory performance, because the assumption that the training and testing data are identically distributed is violated in cross-domain tasks. 2) The semi-supervised method DAELMs (both DAELM-S and DAELM-T) realize some good results due to the labeled data in target domain. And we have proved that the more labeled target samples, the better performance of the algorithm. 3) The traditional DA methods, e.g., TCA, SA, GFK and JDA, perform relatively better, when compared with other supervised and semi-supervised methods. However, they learn a projection matrix to establish the connection between domains, ignoring the inherent discriminative information. 4) The results of all ELM-based methods are comparable, which are slightly less than the proposed ELM-MWMD.

TABLE 4. Average accuracy of the Office and the hand-written digit datasets.

Task/Method	SVM	ELM	TCA	JDA	DAELM-S	DAELM-T	CDELM-M	DST-ELM	ELM-MWMD
A→C	44.08	37.85	**44.52**	43.81	37.11	40.16	42.33	43.72	42.03
A→D	40.76	33.76	33.83	43.95	37.41	45.58	45.86	40.13	**46.50**
A→W	41.69	37.63	41.09	40.00	34.74	42.11	42.85	**44.41**	41.69
C→A	52.40	46.87	**54.77**	51.67	48.21	48.95	52.07	51.88	53.97
C→D	42.04	39.49	46.44	47.13	44.90	46.26	45.86	44.59	**49.04**
C→W	40.68	41.02	41.44	42.37	47.02	47.02	51.05	46.10	**52.20**
D→A	32.36	31.94	37.80	38.41	34.49	33.54	35.72	34.45	**38.83**
D→C	31.43	30.54	**33.86**	30.37	32.52	33.78	30.31	32.06	32.50
D→W	73.56	72.20	**82.45**	76.95	75.09	70.53	81.76	77.97	81.69
W→A	34.13	36.64	36.54	37.79	38.82	38.40	**39.83**	36.22	39.04
W→C	29.92	30.10	33.83	31.61	31.27	34.68	30.31	33.93	**35.98**
W→D	**87.90**	71.97	84.76	85.35	76.87	74.83	81.15	74.52	71.97
MNIST→USPS	54.11	51.39	52.77	65.02	72.91	61.56	62.47	59.83	**89.78**
USPS→MNIST	25.65	30.70	47.19	41.62	52.61	47.39	46.53	43.90	**63.90**
Average	45.05	42.29	47.95	48.29	47.43	47.48	49.15	47.41	**52.80**

TABLE 5. Average accuracy of the COIL20 dataset.

Task/Method	SVM	ELM	TCA	PCA	SA	GFK	JUC-SDELM	ELM-MWMD
COIL1→COIL2	82.22	77.64	85.97	83.06	86.81	85.97	90.28	**92.64**
COIL2→COIL1	81.94	73.75	83.61	82.36	84.58	85.56	86.94	**88.75**
Average	82.08	75.70	84.79	82.71	85.70	85.77	88.61	**90.69**

In fact, the performance of these methods suffer from the number of hidden nodes, and the detailed effect shall be discussed later.

Because we predicted the target data in the iterative process to improve the classification accuracy, the performance varying with the iterations is also displayed in Fig 4. As expected, the accuracies increase iteratively and finally converge after several iterations (All cross-domain tasks were tested and the trends are much the same, so we only select some for display).

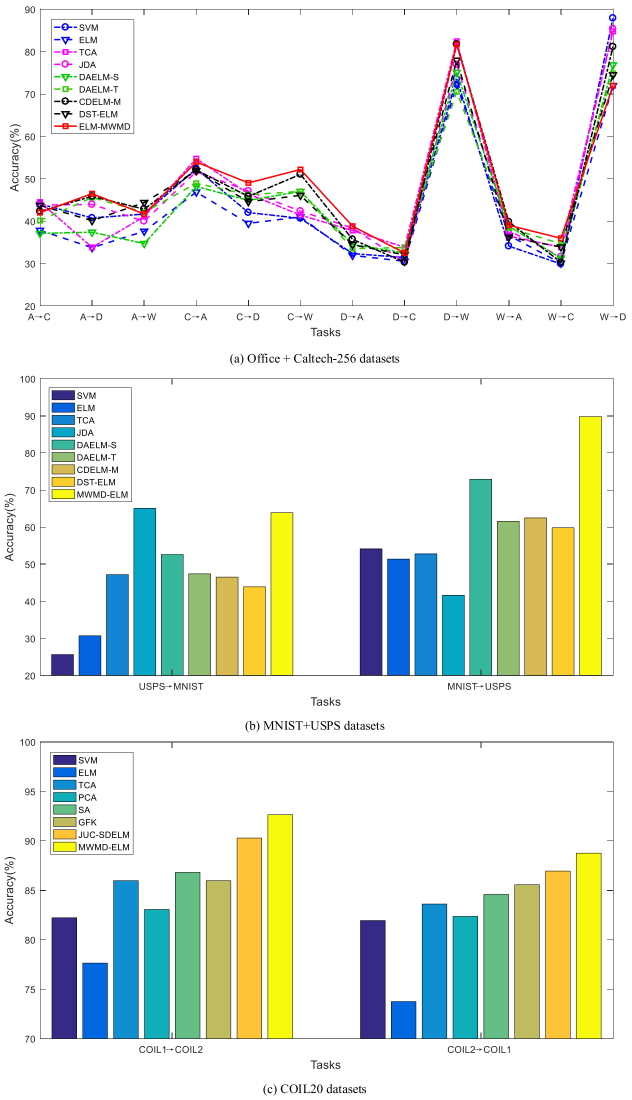
D. PARAMETER SENSITIVITY

In ELM-MWMD, the performance is affected by three tunable parameters: 1) L, the number of the hidden layer neuron; 2) C, the penalty factor in ELM; 3) λ, the tradeoff parameter of MMD. In order to validate the parameter sensitivity, we conducted contrast experiments on different cross-domain tasks, including W→D, MNIST→USPS, USPS→MNIST, COIL1→COIL2 and COIL2→COIL1. The results are displayed in Fig 5.

In Fig 5(a), the influence of the hidden neuron number L is displayed. It can be seen that most of the tasks achieve a good classification accuracy and tend to stabilize with a small scale of the hidden nodes. When the number is less than 500, the classification accuracies improve quickly, and when the number is more than 2000, the accuracies smooth in fluctuation. Although a larger network is helpful for the exploration and exploitation of the invariant features of both domains, a huge number of hidden layer nodes results in heavy computation of the algorithm. Moreover, too many hidden nodes may not be beneficial because they may lead to better output function approximation but degrade the performance on the adaptation, such as MMD measurement.

The penalty factor C balances the regularization and training errors in ELM. It is obvious in Fig 5(b) that the accuracy of varies very little on COIL1→COIL2 and COIL2→COIL1, which indicates the insensitive performance to the value of C. In contrast, the accuracy of the other three tasks varies widely, particularly on task W→D. When the value of C is large, the results of all tasks are reduced. The possible reason is that the weakened regularization term reduces the generalization performance of the model, resulting in overfitting.

The tradeoff parameter λ controls the effect of the distribution adaptation. From Fig 5(c), we can see that the hand-written digits datasets is more sensitive to λ than other



(a) Office + Caltech-256 datasets

(b) MNIST+USPS datasets

(c) COIL20 datasets

FIGURE 3. Accuracy of all tasks on different datasets.

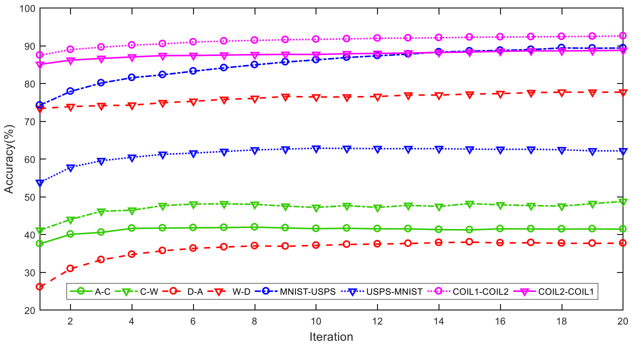
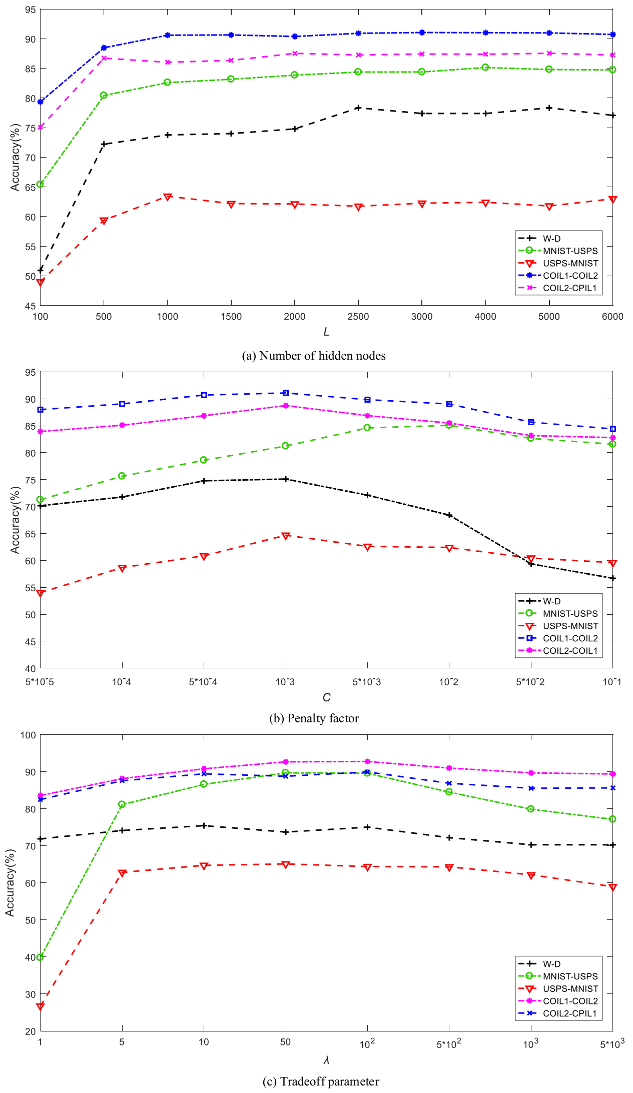


FIGURE 4. Accuracy of all tasks on different datasets.

tasks. Especially when the value of λ is small, the classification accuracy of MNIST→USPS and USPS→MNIST improve rapidly. When λ increases, the results of all tasks become relative stable, which indicates that the projected MMD can effectively minimize the distance between source and target domain.



(a) Number of hidden nodes

(b) Penalty factor

(c) Tradeoff parameter

FIGURE 5. Parameter sensitivity on different tasks.

V. CONCLUSION

In this paper, we present an ELM-MWMD method to deal with the unsupervised domain adaptation problems. Being different from the ordinary methods based on MMD metric, we fully consider the individual information and assign cross-domain weight coefficients to each sample, instead of only considering the global distribution information. Meanwhile, the joint distribution adaptation strategy is utilized to learn an adaptive ELM classifier with labeled source data and unlabeled target data. Thus, our method reserves a simple calculation due to the ELM base, which is beneficial for the learning speed. Besides, the ELM-MWMD can achieve good performance with a small number of hidden nodes, avoiding large-scale calculation. Extensive experiments on real-world image datasets demonstrate that the proposed method outperforms most of the compared baselines.

In the future, we will research further about ELM in unsupervised domain adaptation. For example, the random initialization of ELM affects the stability of the algorithm to some extent, and unselected pseudo-labels may lead to negative transfer, such factors that were not taken into account, will be our next object of research.

REFERENCES

[1] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian, "Unsupervised cross-dataset transfer learning for person re-identificatio," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1306–1315.

[2] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, Apr. 2017.

[3] C. Deng, X. Liu, C. Li, and D. Tao, "Active multi-kernel domain adaptation for hyperspectral image classification," *Pattern Recognit.*, vol. 77, pp. 306–315, May 2018.

[4] S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe, and S. Mougiakakou, "Multi-source transfer learning with convolutional neural networks for lung pattern analysis," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 76–84, Jan. 2017.

[5] F. Mahmood, R. Chen, and N. J. Durr, "Unsupervised reverse domain adaptation for synthetic medical images via adversarial training," *IEEE Trans. Med. Imag.*, vol. 37, no. 12, pp. 2572–2581, Dec. 2018.

[6] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, Nov. 2016, pp. 1568–1575.

[7] S. Sun, H. Liu, J. Meng, C. L. P. Chen, and Y. Yang, "Substructural regularization with data-sensitive granularity for sequence transfer learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2545–2557, Jun. 2018.

[8] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5018–5027.

[9] X. Jia, Y. Jin, X. Su, and Y. Hu, "Domain-invariant representation learning using an unsupervised domain adversarial adaptation deep neural network," *Neurocomputing*, vol. 355, pp. 209–220, Aug. 2019.

[10] 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.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[12] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[13] Y. Peng, S. Wang, X. Long, and B.-L. Lu, "Discriminative graph regularized extreme learning machine and its application to face recognition," *Neurocomputing*, vol. 149, pp. 340–353, Feb. 2015.

[14] T. Guo, L. Zhang, and X. Tan, "Neuron pruning-based discriminative extreme learning machine for pattern classification," *Cognit. Comput.*, vol. 9, no. 4, pp. 581–595, May 2017.

[15] L. Zhang and D. Zhang, "Domain adaptation extreme learning machines for drift compensation in E-Nose systems," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1790–1801, Jul. 2015.

[16] Z. Ma, G. Luo, K. Qin, N. Wang, and W. Niu, "Online sensor drift compensation for E-nose systems using domain adaptation and extreme learning machine," *Sensors*, vol. 18, no. 3, pp. 742–770, Mar. 2018.

[17] S. Xu, X. Mu, D. Chai, and C. Luo, "Domain adaption algorithm with ELM parameter transfer," *Acta Automatica Sinica*, vol. 44, no. 2, pp. 311–317, Feb. 2018.

[18] Y. Liu, L. Zhang, P. Deng, and Z. He, "Common subspace learning via cross-domain extreme learning machine," *Cognit. Comput.*, vol. 9, no. 4, pp. 555–563, May 2017.

[19] 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.

[20] B. Zhang, Y. Liu, H. Yuan, L. Sun, and Z. Ma, "A joint unsupervised cross-domain model via scalable discriminative extreme learning machine," *Cognit. Comput.*, vol. 10, no. 4, pp. 577–590, Apr. 2018.

[21] S. Li, S. Song, G. Huang, and C. Wu, "Cross-domain extreme learning machines for domain adaptation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 6, pp. 1194–1207, Jun. 2019.

[22] X. Jia, M. Zhao, Y. Di, Q. Yang, and J. Lee, "Assessment of data suitability for machine prognosis using maximum mean discrepancy," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5872–5881, Jul. 2018.

[23] M. Long, J. Wang, J. Sun, and P. S. Yu, "Domain invariant transfer kernel learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1519–1532, Jun. 2015.

[24] N. Lu, F. Chu, H. Qi, and S. Xia, "A new domain adaption algorithm based on weights adaption from the source domain," *IEEJ Trans. Electr. Electron. Eng.*, vol. 13, no. 12, pp. 1769–1776, Dec. 2018.

[25] S. Zang, Y. Cheng, X. Wang, Q. Yu, and G.-S. Xie, "Cross domain mean approximation for unsupervised domain adaptation," *IEEE Access*, vol. 8, pp. 139052–139069, 2020.

[26] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," 2019, *arXiv:1903.04687*. [Online]. Available: http://arxiv.org/abs/1903.04687

[27] Z. Liang, L. Zhang, J. Liu, and Y. Zhou, "Adaptively weighted learning for twin support vector machines via bregman divergences," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3323–3336, Apr. 2020.

[28] M. Ponti, J. Kittler, M. Riva, T. D. Campos, and C. Zor, "A decision cognizant Kullback–Leibler divergence," *Pattern Recognit.*, vol. 61, pp. 470–478, Jan. 2017.

[29] 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.

[30] 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.*, Sydney, NSW, Australia, Dec. 2013, pp. 2200–2207.

[31] K. Saenko, B. J. Kulis, M. Fritz, and T. J. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.* Crete, Greece: Springer-Verlag, Sep. 2010, pp. 213–226.

[32] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*. Pasadena, CA, USA: California Institute of Technology, 2007.

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

[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

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

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

YANNA SI received the B.E. degree from the School of Information Engineering, Henan University of Science and Technology, Luoyang, China, in 2014, where she is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems. Her current research interests include intelligent control, reinforcement learning, and transfer learning.

JIEXIN PU received the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, China, in 2007. He is currently a Doctoral Supervisor with the School of Information Engineering, Henan University of Science and Technology, China. His current research interests include intelligent control, pattern recognition, and computer vision.

SHAOFEI ZANG received the Ph.D. degree in control science and control engineering from the China University of Mining and Technology, Xuzhou, China, in 2017. He is currently an Assistant Professor with the Department of Information Engineering, Henan University of Science and Technology. His research interests include machine learning and computer vision.

LIFAN SUN received the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2014. He is currently an Associate Professor with the School of Information Engineering, Henan University of Science and Technology, Henan, China. His research interests include information fusion, object tracking, signal and data processing, optimal estimation, artificial intelligence, and performance evaluation.