Flexible Label-Induced Manifold Broad Learning System for Multiclass Recognition

Junwei Jin, Biao Geng, Yanting Li, Jing Liang, *Senior Member, IEEE*,
Yang Xiao, *Fellow, IEEE*, and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*— Broad learning system (BLS), which emerges as a lightweight network paradigm, has recently attracted great attention for recognition problems due to its good balance between efficiency and accuracy. However, the supervision mechanism in BLS and its variants generally relies on the strict binary label matrix, which imposes limitations on approximation and fails to adequately align with the data distribution. To address this issue, in this article, two novel flexible label-induced BLS models with the manifold manner are proposed, whose notable characteristics are as follows. First, two proposed label relaxation strategies can both enlarge the margins between different categories and simultaneously enhance the diversity within labels. Second, the integration of manifold geometrical criterion enables the models to capture local feature structures, ensuring the obtained flexible labels align better with the similarity between samples. Third, the proposed models can be optimized efficiently with the alternating direction method of multipliers. Each iteration benefits from a closed-form solution, facilitating the optimization process. Extensive experiments and thorough theoretical analysis are intended to show the advantages of our proposed models compared to other state-of-the-art recognition algorithms.**

***Index Terms*— Broad learning system (BLS), classification, label dragging, manifold geometry, retargeted.**

Nomenclature

$\boldsymbol{X}$	Data matrix.
$\boldsymbol{Y}$	Label matrix.
$\boldsymbol{L}$	Graph Laplacian matrix.
$\boldsymbol{W}$	Output weights of BLS.
$\boldsymbol{B}$	Luxury matrix.
$\boldsymbol{M}$	Nonnegative relaxation matrix.
$\boldsymbol{T}$	Marginalized retargeted matrix.
$\boldsymbol{A}$	Broad transformed feature in BLS.
$\boldsymbol{Z}^{N_g}$	Groups of random features in BLS.
N_g	Number of groups in $\boldsymbol{Z}^{N_g}$.
N_f	Feature nodes in each group of $\boldsymbol{Z}^{N_g}$.
N_e	Number of enhancement nodes $\boldsymbol{H}^m$.
d	Dimension of the broad feature $\boldsymbol{A}$.

I. Introduction

OVER the past decade, a tremendous amount of neural network-based methods have been developed to tackle multiclass recognition tasks, encompassing diverse applications ranging from face recognition [1], medical image analysis, action recognition [2] to remote sensing image classification [3]. The deep network is a typical learning paradigm which equips with multiple hidden layers to extract more abstract features [4]. Supported by high-performance hardware, deep models have achieved vigorous progress in the modern big data era [5]. Nevertheless, the fact that deep networks typically involve numerous hyperparameters that are optimized through time-consuming gradient descent and are difficult to be analyzed theoretically, makes it imperative to design more effective neural network algorithms [6].

Recently, a randomized neural network called broad learning system (BLS) [7] has garnered significant attention in the artificial intelligence community. Fig. 1 presents the diagram of a standard BLS. The inputs are first mapped to generate a series of random features, which are fine-tuned by the sparse autoencoder. Then, using random projection, these random features are propagated to another set of enhanced features. Finally, these two types of features are concatenated and connected to the label space, where the weights can be derived by solving a ridge regression analytically [8]. Its specific execution modules we refer to Section II-B. The prominent advantages of BLS are its concise network architecture and computational attractiveness, which have led to the emergence of numerous variants. To name a few, Chen et al. [9]

Manuscript received 27 November 2022; revised 2 June 2023; accepted 29 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62106068, Grant 62106233, Grant U1801262, Grant U1813203, and Grant 61751205; in part by the Science and Technology Research Project of Henan Province under Grant 212102210148 and Grant 222102210058; and in part by the Grant 22ZZRDZX29, Grant KFJJ2022001, and Grant 2021ZKCJ14. *(Corresponding authors: Yanting Li; C. L. Philip Chen.)*

Junwei Jin and Biao Geng are with the Key Laboratory of Grain Information Processing and Control, Ministry of Education, Henan University of Technology, Zhengzhou 450001, China, also with the Henan Provincial Key Laboratory of Grain Photoelectric Detection and Control, Zhengzhou 450001, China, and also with the School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou 450001, China (e-mail: jinjunwei24@163.com; gengbiao2002@163.com).

Yanting Li is with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China (e-mail: ytli1227@126.com).

Jing Liang is with the School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: liangjing@zzu.edu.cn).

Yang Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: yangxiao@ieee.org).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: hilip.chen@ieee.org).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3291793.

Digital Object Identifier 10.1109/TNNLS.2023.3291793

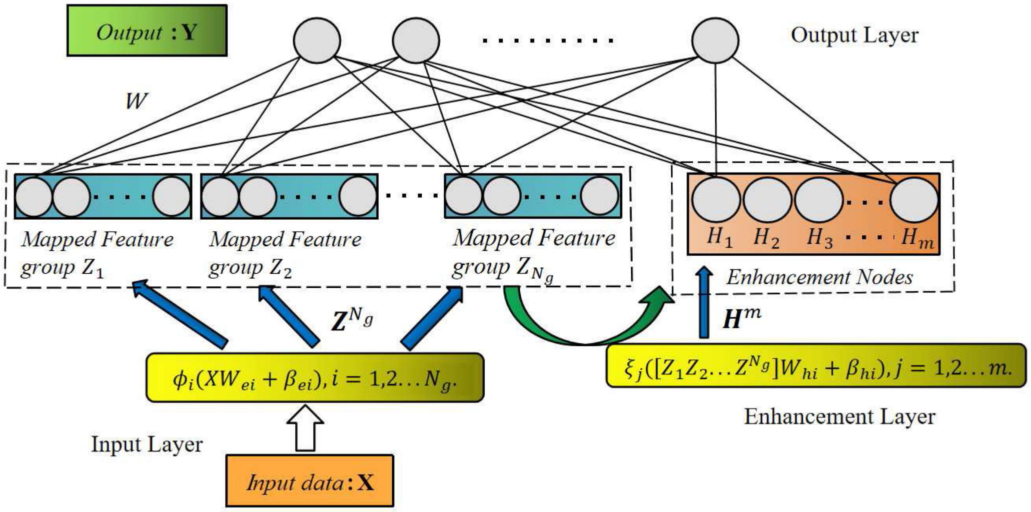

Output : Y
Output Layer
W
Mapped Feature group Z_1
Mapped Feature group Z_2
Mapped Feature group Z_{N_g}
H_1 H_2 H_3 $\cdots$ H_m
Enhancement Nodes
Z^{N_g}
H^m
$\phi_i(XW_{ei}+\beta_{ei}), i = 1,2\ldots N_g.$
$\xi_j([Z_1Z_2\ldots Z^{N_g}]W_{hi}+\beta_{hi}), j = 1,2\ldots m.$
Enhancement Layer
Input Layer
Input data : X

Fig. 1. Network structure of the standard BLS.

demonstrated the universal approximation ability of BLS, and several other broad architectures with their mathematical derivations were also presented [10]. Wu et al. [11] further consolidated BLS's theoretical basis from the perspective of fractional calculus. Then, by extracting random features with a separate TS fuzzy subsystem, Feng et al. [12] proposed a fuzzy broad learning algorithm for classification and regression. Jin et al. [13] even designed a maximum likelihood robust objective function for BLS to recognize patterns with corrupted labeling. More summary about BLS can be found in [14].

In most cases, BLS and its variants can achieve satisfactory performances for multiclass recognition tasks. However, the target vector they utilized is generally strictly labeled, i.e., only its jth entry is 1 following the instance's class ownership, and the others are all 0. In practice, such binary labels cannot be optimal for supervision. First, it restricts the flexibility of projection learning due to its strictness. Second, it cannot effectively separate samples of different classes as the distances between the correct and incorrect label vectors remain constant (i.e., $\sqrt{2}$). What is more, the underlying locally invariant property of data has not been fully considered in the modeling. As a result, the generalization performances of broad networks can be severely restricted.

To address these deficiencies, many researchers have paid attention to adjust the labeling strategy and manifold learning. To enhance the flexibility of supervised targets, Xiang et al. [15] proposed a ε-dragging technology to relax the discrete label matrix, and the regression labels of different classes can be forced to move in opposite directions. Building upon this idea, Zhang et al. [16] proposed to directly learn the transformed regression with a large margin constraint during model training. Later, Wang and Pan [17] pointed out that the difference between the two soft labels in [15] and [16] can be represented as a translation vector, which was further constrained by a groupwise regularizer to improve the recognition performance. Moreover, Geng [18] proposed a continuous numerical label which can provide richer semantic information for recognition. To discover the geometrical structure of data, Han et al. [19] proposed a label-based graph that imposes strict constraints, forcing samples of the same class to be grouped together. Zhang et al. [20] leveraged unlabeled data in conjunction with labeled data to develop a neighborhood-preserving local nonlinear manifold feature extractor. Du et al. [21] even considered a label enhancement method with a graph manner for the class-imbalance learning. Although these flexible label-based methods show remarkable performance in classification, most of them are directly applied to linear models using the original feature space. This can lead to a decline in recognition performance due to the lack of a discriminative feature transformation.

In light of the aforementioned observations, novel manifold BLS models with flexible label will be proposed in this article. As the strict binary label matrix cannot fully excavate the rich characteristics and variability within different data, our priority is to readjust the regression targets in BLS with two novel proposed relaxed strategies, i.e., label dragging and supervised retargeted. The first one emphasizes the expression of soft label movement explicitly, while adaptive target matrix learning is focused on by the second method. Nonetheless, their common ultimate effects are to enlarge the distances between different categories and provide more judicious supervised information for recognition. Furthermore, we incorporate the manifold geometrical criterion to build a class compactness graph to capture the local feature structures, which can guide the flexible labels to be better consistent with the distribution of data. By doing so, a more discriminative output weights can be trained under the double supervision of the optimal slack variable targets and manifold criterion, which will undoubtedly facilitate the classifiers' generalization performances. To the end, efficient and effective iterative algorithms are derived to optimize the proposed models, whose convergence and complexity are also guaranteed by thorough theoretical analysis. Extensive experiments on diverse image databases are conducted to validate the distinguished performances of our proposed methods. In summary, the contributions of this article can be listed as follows.

1) Two flexible label learning mechanisms are proposed and for the first time imposed on the framework of BLS with a manifold manner, which can make the margins between different categories more suitable for classification.
2) Two effective optimization algorithms are developed to solve the resulting models, and solid theoretical analysis and experimental results are presented to ensure their computational efficiency and convergence well.
3) Diverse experiments are carried out to verify the effectiveness of our proposed methods in comparison with many other state-of-the-art recognition algorithms.

The rest of this article is organized as follows. Section II introduces several preliminaries. The details of our proposed two models are presented in Section III. In Section IV, we give the iterative optimization algorithms and theoretical analysis. Section V presents the experimental results obtained from various multiclass recognition tasks. Finally, the conclusions and future work are summarized in Section VI.

II. Preliminaries

A. Notation

In this article, $\boldsymbol{I}$ means the identity matrix, the ith row and jth column of the matrix $\boldsymbol{\Omega}$ is represented as $\boldsymbol{\Omega}(i,:)$ and $\boldsymbol{\Omega}(:,j)$. $\|\boldsymbol{\Omega}\|_F^2 = \mathrm{tr}(\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\Omega}) = \mathrm{tr}(\boldsymbol{\Omega}\boldsymbol{\Omega}^{\mathrm{T}})$ denotes $\boldsymbol{\Omega}$'s Frobenius norm, where $\boldsymbol{\Omega}^{\mathrm{T}}$ is its transpose and $\mathrm{tr}(\cdot)$ denotes the trace operator. $\{\boldsymbol{X} \in \mathbb{R}^{N\times d_x}, \boldsymbol{Y} \in \mathbb{R}^{N\times C}\}$ denotes the training data from C classes, where N and d_x, respectively, denote the number and dimension of train samples. In $\boldsymbol{Y}$'s kth row $\boldsymbol{Y}(k,:)$, only the ith entry is 1 if sample x_k comes from the ith class, the remaining entries are all 0. $\|\cdot\|_2$ denotes the ℓ_2 norm of a vector, $\boldsymbol{\Omega}^{-1}$ denotes the inverse of matrix, and $\odot$ is the Hadamard product operator (multiply elementwisely). We list the interpretations of some commonly used notations in Nomenclature, other temporary symbols will be specified when they are encountered.

B. Standard BLS Model

Given training data $\{\boldsymbol{X}, \boldsymbol{Y}\}$, the specific execution modules of the standard BLS model can be summarized as follows.

1) Random Feature Module: Randomly generate weights $\boldsymbol{W}_{e_i}$ and biases $\boldsymbol{\beta}_{e_i}$, and then N_g groups of mapping features from the input data $\boldsymbol{X}$ can be extracted with the transformation mapping function $\boldsymbol{\phi_i}(\cdot)$. The ith group of mapping features with N_f nodes can be expressed as

$$\boldsymbol{Z}_i = \boldsymbol{\phi}_i\left(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i}\right), \quad i = 1, 2, \ldots, N_g. \tag{1}$$

Then, concatenating all these groups of mapping features, we can get the first kind of random feature as

$$\boldsymbol{Z}^{N_g} \triangleq \left[\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_{N_g}\right] \in \mathbb{R}^{N\times(N_g\times N_f)}. \tag{2}$$

Here it should be noted that to extract more compact and discriminative feature $\boldsymbol{Z}_i$, we can adopt the sparse autoencoder to fine-tune the weights $\boldsymbol{W}_{e_i}$ and bias $\boldsymbol{\beta}_{e_i}$.

2) Enhanced Feature Module: After getting the random feature $\boldsymbol{Z}^{N_g}$, we can map it into a so-called enhancement layer with the weight $\boldsymbol{W}_{h_j}$ and bias $\boldsymbol{\beta}_{h_j}$, and the jth enhancement group can be obtained as follows:

$$\boldsymbol{H}_j = \boldsymbol{\xi}_j\left(\boldsymbol{Z}^{N_g}\boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}\right), \quad j = 1, 2, \ldots, m \tag{3}$$

where $\boldsymbol{\xi}(\cdot)$ represents the activation function, and the weight $\boldsymbol{W}_{h_j}$ and bias $\boldsymbol{\beta}_{h_j}$ are still randomly generated. We can see this part plays the similar role as the hidden layer in general networks.

Then, combining all these enhancement groups, we can get the second of enhanced feature like

$$\boldsymbol{H}^m \triangleq [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m] \in \mathbb{R}^{N\times N_e}. \tag{4}$$

3) Output Layer Module: Based on the obtained $\boldsymbol{Z}^{N_g}$ and $\boldsymbol{H}^m$, we can get the broad transformed feature $\boldsymbol{A}$ like

$$\boldsymbol{A} \triangleq \left[\boldsymbol{Z}^{N_g}, \boldsymbol{H}^m\right] \in \mathbb{R}^{N\times(N_g\times N_f+N_e)}. \tag{5}$$

Up to now, the obtained broad feature can be transformed to the predefined target matrix $\boldsymbol{Y}$ with a transformation matrix $\boldsymbol{W}$. Then, the key issue is how to solve $\boldsymbol{W}$ discriminatively. Different from the previous network weights which are obtained randomly, here $\boldsymbol{W}$ can be attained by optimizing a least square problem as follows:

$$\min_{\boldsymbol{W}} \|\boldsymbol{A}\boldsymbol{W} - \boldsymbol{Y}\|_F^2 + \lambda\|\boldsymbol{W}\|_F^2 \tag{6}$$

where the first term can measure the approximation error between the ideal labels and the predicted ones, the second term can smooth the distribution of $\boldsymbol{W}$ to avoid overfitting, and λ is a regularization parameter to balance the importance of these two terms. In general, (6) is called as the objective function of the BLS model, which can be solved by setting its derivation with respect to $\boldsymbol{W}$ to be 0. Thus, we can get

$$\boldsymbol{W} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y} \tag{7}$$

where $\boldsymbol{A}^{+} = \lim_{\lambda\to 0}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}$ is just the Moore–Penrose generalized inverse of $\boldsymbol{A}$.

III. Proposed Models

To ensure a logical flow of the narration, this section begins by presenting a general framework comprising a flexible target cost function and a manifold regularizer. Subsequently, two distinct flexible label strategies are proposed according to this framework. Finally, the rationality analysis on the modeling will be provided.

A. General Framework

From the preceding descriptions of the standard BLS, it is evident that its success can be primarily attributed to its powerful feature acquisition and weight calculation mechanisms. However, the utilized strict binary label space not only provides little freedom for the regression process but also fails to maintain the locally invariant property of data. To this end, we propose to learn flexible labels within the manifold latent space, and a general framework of the derived methods can be formulated as

$$\min_{\boldsymbol{W}} \|\boldsymbol{A}\boldsymbol{W} - \Delta\boldsymbol{Y}\|_F^2 + \lambda_1 E_M + \lambda_2 \|\boldsymbol{W}\|_F^2 \tag{8}$$

where $\boldsymbol{A}$ denotes the broad feature, λ_1 and λ_2 are two trade-off regularization parameters.

Comparing (6) and (8), their main difference lies in the introduction of $\Delta\boldsymbol{Y}$ and E_M. In our intention, $\Delta\boldsymbol{Y}$ denotes the flexible version of the binary target $\boldsymbol{Y}$ which needs to be designed, and E_M represents the manifold regularizer which aims to preserve the similarity between data during transformation. To achieve this purpose, a class compactness undirected graph should be first built to link the samples from the same class [22], and the adjacency matrix can be defined as

$$\boldsymbol{V}_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}}, & \text{if } \operatorname{label}(x_i) = \operatorname{label}(x_j) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where σ is the bandwidth heat kernel parameter. Based on the key idea of manifold learning, the similarity between data in the transformed space can be preserved well by minimizing the following function:

$$\sum_{ij} \|\boldsymbol{A}_i\boldsymbol{W} - \boldsymbol{A}_j\boldsymbol{W}\|_2^2 \boldsymbol{V}_{ij} = \operatorname{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A}\boldsymbol{W}\right) \tag{10}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{V}$ is the graph Laplacian, and $\boldsymbol{D}$ represents a diagonal matrix with $\boldsymbol{D}_{ii} = \Sigma_j \boldsymbol{V}_{ij}$.

It is by now clear that minimizing (10) with appreciate weight $\boldsymbol{V}_{ij}$ can ensure the transformed samples from the same class be kept close to each other. Thus, replacing E_M in (8) with (10), we can get the following framework formulation:

$$\min_{\boldsymbol{W}} \|\boldsymbol{A}\boldsymbol{W} - \Delta\boldsymbol{Y}\|_F^2 + \lambda_1 \operatorname{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A}\boldsymbol{W}\right) + \lambda_2 \|\boldsymbol{W}\|_F^2. \tag{11}$$

Note that in (11), the commonly utilized binary labeling strategy fixes the distance of different categories as $\sqrt{2}$, which deviates from the manifold assumption. To address this issue, we will propose two novel approaches for label relaxation in Sections III-B and III-C.

B. LDMBLS

In this part, a label dragging technique is adopted to relax the strict binary targets into the slack but discriminative ones. Foremost, we present an example to illustrate the principle of the label dragging technique which can also explain the strong discriminability of the slack labels compared with strict targets for classification. Choosing three samples, respectively, from the second, third, and first class, and their zero-one label matrix can be expressed as

$$\boldsymbol{Y} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}.$$

Obviously, the margins between different classes are fixed as $\sqrt{2}$, which cannot reflect well the similarity between different samples. From the viewpoint of manifold learning, the transformed samples from different categories should be as far as possible. Thus, this strict binary strategy in labeling needs to be relaxed into the flexible constraints to provide more freedom for training the learner. Our approach is to drag these binary labels far away in opposite directions. As a result, the binary $\boldsymbol{Y}$ can be transformed to a slack variable version

$$\Delta\boldsymbol{Y} = \begin{bmatrix} -\varepsilon_{11} & 1+\varepsilon_{12} & -\varepsilon_{13} \\ -\varepsilon_{21} & -\varepsilon_{22} & 1+\varepsilon_{23} \\ 1+\varepsilon_{31} & -\varepsilon_{32} & -\varepsilon_{33} \end{bmatrix}, \quad \text{s.t. } \varepsilon_{ij} \geq 0. \tag{12}$$

We can find the nonnegative constraint of εs can help to enlarge the margins between different classes. Take the second and third sample in (12) as an example, whose margin is

$$\begin{aligned} &\sqrt{(-\varepsilon_{21} - 1 - \varepsilon_{31})^2 + (-\varepsilon_{22} + \varepsilon_{32})^2 + (1 + \varepsilon_{23} + \varepsilon_{33})^2} \\ &\geq \sqrt{(0-1)^2 + (1-0)^2 + (0-0)^2}. \end{aligned} \tag{13}$$

Hence, the slack label matrix can better express the correlations between different data, naturally beneficial for learning a discriminative transformation matrix for recognition.

Next, we consider to integrate this effective label dragging technique into our proposed framework (11). To do this, a nonnegative relaxation matrix $\boldsymbol{M}$ is first defined as

$$\boldsymbol{M} = \begin{bmatrix} \varepsilon_{11} & \cdots & \varepsilon_{1c} \\ \vdots & \varepsilon_{i,j} & \vdots \\ \varepsilon_{n1} & \cdots & \varepsilon_{nc} \end{bmatrix}_{N\times C} \geq \boldsymbol{0} \tag{14}$$

which indicates the distances of dragging. Then, we devise a luxury matrix $\boldsymbol{B} \in \mathbb{R}^{N\times C}$ to induce the direction of label movement, whose element is defined as

$$\boldsymbol{B}_{ij} = \begin{cases} +1, & \text{if } \ y_{ij} = 1 \\ -1, & \text{if } \ y_{ij} = 0 \end{cases} \tag{15}$$

where $+1$ and -1 represents the positive and negative directions of label movement, respectively. Combining with the original label matrix $\boldsymbol{Y}$, the slack variable matrix $\Delta\boldsymbol{Y}$ can be expressed as $\boldsymbol{Y} + \boldsymbol{B} \odot \boldsymbol{M}$. Thus, the proposed label dragging-based manifold BLS (LDMBLS) model can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{M}} \ & \|\boldsymbol{A}\boldsymbol{W} - (\boldsymbol{Y} + \boldsymbol{B} \odot \boldsymbol{M})\|_F^2 + \lambda_1 \operatorname{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A}\boldsymbol{W}\right) \\ & + \lambda_2 \|\boldsymbol{W}\|_F^2, \quad \text{s.t.} \quad \boldsymbol{M} \geq \boldsymbol{0}. \end{aligned} \tag{16}$$

C. REMBLS

Rethinking the label dragging technique in (16), we can find that the flexible label space of LDMBLS is handicapped seriously by the bound that the fitted targets are smaller than 0 for false categories and larger than 1 for true categories. The main reason states that the dragging direction of labels is predefined, and the obtained label space cannot be consistent with the distribution of data. In other words, the proposed model is not completely free from the negative influence of the original binary hard label matrix $\boldsymbol{Y}$. Thus, the flexibility and discriminability of the regression model are confined greatly.

To solve this drawback, a retargeted strategy is proposed to relax the label matrix in this part. Specifically, instead of specifying the direction and distances of dragging, the flexible targets $\Delta\boldsymbol{Y}$ should be learned directly

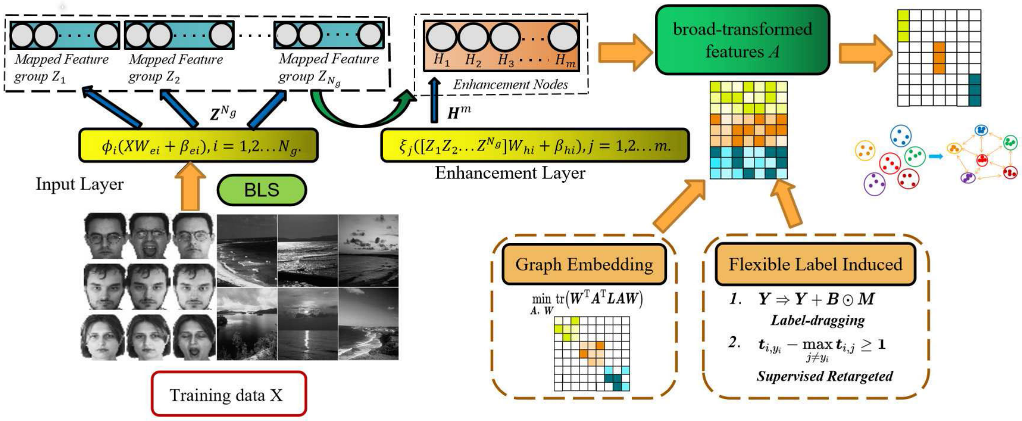

Mapped Feature group Z_1
Mapped Feature group Z_2
Mapped Feature group Z_{N_g}
H_1 H_2 H_3 $\cdots$ H_m
Enhancement Nodes
broad-transformed features A
Z^{N_g}
H^m
$\phi_i(XW_{ei}+\beta_{ei}), i=1,2\ldots N_g.$
$\xi_j([Z_1Z_2\ldots Z^{N_g}]W_{hi}+\beta_{hi}), j=1,2\ldots m.$
Input Layer
Enhancement Layer
BLS
Training data X
Graph Embedding
$\min_{A,W} \mathrm{tr}(W^TA^TLAW)$
Flexible Label Induced
1. $Y \Rightarrow Y + B \odot M$
Label-dragging
2. $t_{i,y_i} - \max_{j\neq y_i} t_{i,j} \geq 1$
Supervised Retargeted

Fig. 2. Flowchart of the proposed flexible label-induced BLS models, including the broad network, flexible label strategy, and graph embedding.

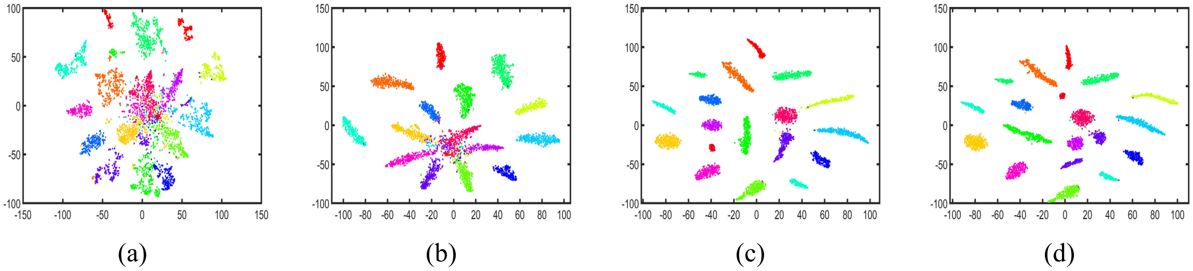


Fig. 3. T-SNE visualization of the feature by (a) original data, (b) BLS, (c) LDMBLS, and (d) REMBLS on the Fifteen Scenes database.

from data, and a marginalized constraint is adopted to make the learned labels distinguishable. Integrating this retargeted scheme into our proposed framework (11), a retargeted manifold BLS (REMBLS) model is presented with the following formulation:

$$\min_{W,T} \|AW - T\|_F^2 + \lambda_1 \operatorname{tr}\big(W^{\mathrm{T}} A^{\mathrm{T}} L A W\big) + \lambda_2 \|W\|_F^2$$
$$\text{s.t. } t_{i,y_i} - \max_{j\neq y_i} t_{i,j} \geq c, \quad i = 1, 2, \ldots, N \tag{17}$$

where A denotes the broad feature, $T = [t_1, t_2, \ldots, t_N]^{\mathbf{T}} \in \mathbb{R}^{N\times C}$ represents the flexible labels which should be learned in the optimization process. Herein the $t_{i,y_i} - \max_{j\neq y_i} t_{i,j} \geq c$ is just the marginalized constraint between true and false class, in which y_i is the true class-indicator of sample x_i. For the sample x_i from the kth category, our key idea of the marginalized constraint term is to force its margin between the kth and other elements in the learned target space by the fixed constant value c. This gentle constraint can not only guide the distribution of learned flexible label space to coincide with the latent manifold space, but also provide more freedom for regression process. For the convenience of calculation and implementation, here the marginal value c between the false and true categories is set to 1.

Apparently, by jointly optimizing W and T with marginalized constraints in (17), the intraclass compactness and interclass separability can be preserved simultaneously. Thus, our proposed REMBLS can be more flexible and discriminative for classification tasks.

D. Rationality Analysis

From the descriptions above, we can see the proposed models offer a range of rich and compact features extracted from the inputs through autoencoders. Then, the feature space can be further enhanced through random projection. To guide an effective connection between the composed broad features and the output targets, we present two different flexible strategies to slack the hard label matrix. Leveraging the principle of manifold learning, a class compactness graph is constructed to enforce the learned soft labels to obey the distribution of source data. As a result, the interclass separability and intraclass compactness within the transformed space can be promoted simultaneously, which indeed benefits the subsequent classification tasks. The flowchart of our proposed BLS-based models is displayed in Fig. 2. To better verify the effective supervision of the soft labels, we employ the t-SNE method to visualize the transformed data of the proposed LDMBLS and REMBLS models. Here the fifteen scene category database with 15 tr per class is utilized as the benchmark, and all samples' transformed data in LDMBLS and REMBLS are embedded into two dimensions to be visualized. For comparison, the standard BLS model is also trained, and its transformed feature as well as the original data are processed as above. The visualization results are demonstrated in Fig. 3. Obviously, we can see that the original features are cluttered with mixed structures, which is not conducive to classification. Through mapping and composing features broadly in the standard BLS model, it can be found that the feature space has been compact and separable. Nonetheless, many sample points from different categories are still mixed with each other.

Apparently, the transformed data by our proposed LDMBLS and REMBLS can alleviate these shortcomings well. As illustrated in Fig. 3(c) and (d), the data from the same class exhibits clear clusters with highly separable representations. From these, we can conclude that the novel merged flexible label and manifold terms can work together well to guide the margins between different samples to be suitable for classification. Thus, the outstanding performances of our proposed models can be guaranteed well. And further verification results on more databases will be given in Section V.

IV. Optimization and Algorithm Analysis

In this section, two effective iterative algorithms based on the alternating direction multiplier method (ADMM) are first proposed to solve our proposed models (16) and (17), respectively. Then, the convergence and computation complexity of the optimization process will be discussed.

A. Optimization of LDMBLS

As there are two unknown variables $\boldsymbol{W}$ and $\boldsymbol{M}$ in the objective function of LDMBLS, problem (16) cannot be optimized directly with an analytical solution. Therefore, here we resort to the alternating direction method, i.e., minimizing the subproblems with respect to one variable by fixing the other one. Specifically, this minimization procedure can be expanded into two steps, whose details are as follows.

1) Optimizing $\boldsymbol{W}$: Fix the variable $\boldsymbol{M}$, the optimization of $\boldsymbol{W}$ can be obtained by minimizing the following function:

$$\Psi(\boldsymbol{W}) = \|\boldsymbol{A}\boldsymbol{W} - \boldsymbol{S}\|_F^2 + \lambda_1 \operatorname{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A}\boldsymbol{W}\right) + \lambda_2\|\boldsymbol{W}\|_F^2 \tag{18}$$

where $\boldsymbol{S} = \boldsymbol{Y} + \boldsymbol{B} \odot \boldsymbol{M}$. The resulting function (18) is a typical regularized least square problem, thus, it can be solved directly by setting its derivation with respect to $\boldsymbol{W}$ to be 0, i.e.,

$$\begin{aligned}\frac{\partial \Psi(\boldsymbol{W})}{\partial \boldsymbol{W}} &= \boldsymbol{0}\\ &\Rightarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{W} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{S} + \lambda_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A}\boldsymbol{W} + \lambda_2\boldsymbol{W} = \boldsymbol{0}\\ &\Rightarrow \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A} + \lambda_2\boldsymbol{I}\right)\boldsymbol{W} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}.\end{aligned} \tag{19}$$

Thus, the optimal $\boldsymbol{W}$ in (16) can be expressed as

$$\boldsymbol{W} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A} + \lambda_2\boldsymbol{I}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}. \tag{20}$$

2) Optimizing $\boldsymbol{M}$: Fix the optimized variable $\boldsymbol{W}$ and ignore the terms irrelevant to $\boldsymbol{M}$, the minimization problem (16) can be rewritten as

$$\min_{\boldsymbol{M}} \|\boldsymbol{Q} - \boldsymbol{B} \odot \boldsymbol{M}\|_F^2, \quad \text{s.t.} \quad \boldsymbol{M} \geq 0 \tag{21}$$

where $\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{W} - \boldsymbol{Y}$. Taking account into the property of the squared Frobenius norm of matrix, problem (21) can be decomposed into $N \times C$ subproblems which can be optimized one by one. For $\boldsymbol{M}_{ij}$, the derived subproblem is just like

$$\min_{\boldsymbol{M}_{ij}}\left(\boldsymbol{Q}_{ij} - \boldsymbol{B}_{ij}\boldsymbol{M}_{ij}\right)^2, \quad \text{s.t.} \quad \boldsymbol{M}_{ij} \geq 0 \tag{22}$$

which is a simple unary quadratic problem. According to the construction of the luxury matrix $\boldsymbol{B}$, we know that $\boldsymbol{B}_{ij}^2 = 1$. Thus, (22) can be converted to

$$\left(\boldsymbol{Q}_{ij} - \boldsymbol{B}_{ij}\boldsymbol{M}_{ij}\right)^2 = \left(\boldsymbol{B}_{ij}\boldsymbol{Q}_{ij} - \boldsymbol{M}_{ij}\right)^2. \tag{23}$$

Consider the nonnegative constraint of entry $\boldsymbol{M}_{ij}$, problem (22) can be optimized with a solution as

$$\boldsymbol{M}_{ij} = \max\left(\boldsymbol{B}_{ij}\boldsymbol{Q}_{ij}, 0\right). \tag{24}$$

As a result, we can get the optimal $\boldsymbol{M}$ by concatenating all the $\boldsymbol{M}_{ij}$ together, and a compact matrix form can be presented as

$$\boldsymbol{M} = \max(\boldsymbol{B} \odot \boldsymbol{Q}, \boldsymbol{0}). \tag{25}$$

In implementation, (20) and (25) can be recursively repeated until the difference between two iterations is less than a predetermined threshold. To give readers an overall understanding of this iteration procedure, we outline the details to solve (16) in Algorithm 1.

Algorithm 1 Optimizing (16) by ADMM

```
Require: Broad transformed feature $\boldsymbol{A}$; binary label matrix
    $\boldsymbol{Y}$; the regularization parameters $\lambda_1$ and $\lambda_2$;
    Initialization: $\boldsymbol{M} = \boldsymbol{0}$;
    1. Calculate the Laplacian matrix $\boldsymbol{L}$ with Eq. (9);
    2. Construct the luxury direction matrix $\boldsymbol{B}$ with Eq. (15);

    Set iteration number $t = 1$;
    while not converged do
        3. Optimize $\boldsymbol{W}$ with Eq. (20);
        4. Optimize $\boldsymbol{M}$ with Eq. (25);
        5. Update iteration number $t = t + 1$;
    end while
Ensure: Output weight matrix $\boldsymbol{W}$.
```

B. Optimization of REMBLS

Similar to (16), problem (17) still needs to be optimized by the framework of alternative direction method. First, we focus on the updating of retargeted matrix $\boldsymbol{T}$. Given $\boldsymbol{W}$ and ignoring the terms irrelevant to $\boldsymbol{T}$ in (17), we can get the following minimization equation as

$$\min_{\boldsymbol{T}} \|\boldsymbol{P} - \boldsymbol{T}\|_F^2, \ \text{s.t.} \ \boldsymbol{t}_{i,y_i} - \max_{j \neq y_i} \boldsymbol{t}_{i,j} \geq c, \quad i = 1, \ldots, N \tag{26}$$

where $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{W} \in \mathbb{R}^{N \times C}$. Obviously, (26) is a typical quadratic programming with constraints. We should divide (26) into N independent subproblems, and the ith subproblem can be written as

$$\min_{\boldsymbol{t}_i} \|\boldsymbol{p}_i - \boldsymbol{t}_i\|_2^2, \quad \text{s.t.} \quad \boldsymbol{t}_{ik} - \max_{j \neq k} \boldsymbol{t}_{ij} \geq c \tag{27}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{t}_i$ denote the ith row of $\boldsymbol{P}$ and $\boldsymbol{T}$, respectively. Apparently, we have $\|\boldsymbol{p}_i - \boldsymbol{t}_i\|_2^2 = \sum_{j=1}^{c}(p_{ij} - t_{ij})^2$. From the principle of retargeted strategy, we know for the true class $j = k$, $\boldsymbol{t}_{ik} = \boldsymbol{p}_{ik} + \boldsymbol{\rho}$, i.e., $\boldsymbol{p}_{ik}$ can be modified slightly to achieve the optimal value of the true class $\boldsymbol{t}_{ik}$. And for the false class $\forall j \neq k$, we have $\boldsymbol{t}_{ik} - \boldsymbol{t}_{ij} \geq c$. Thus, problem

Algorithm 2 Optimizing (26)

```
Require: Broad transformed feature $\boldsymbol{A}$; the true class index $k$;
    Initialization: $\boldsymbol{\kappa}_j = \boldsymbol{t}_{ij}+1-\boldsymbol{t}_{ik}$; $\boldsymbol{\rho} = 0$; $\boldsymbol{W} = \boldsymbol{0}$; $iter = 0$;
    1. For $j = 1 : C \quad (j \neq k)$
    2.     if $\Phi'(\boldsymbol{\kappa}_j) = \boldsymbol{\kappa}_j + \sum_{q \neq k} \min(\boldsymbol{\kappa}_j - \boldsymbol{\kappa}_q, 0) > 0$
    3.         set $\boldsymbol{\rho} = \boldsymbol{\rho} + \boldsymbol{\kappa}_j$ and $iter = iter + 1$;
    4.     end if
    5. end for
    6. Set $\boldsymbol{\rho} = \frac{\boldsymbol{\rho}}{1+iter}$;
    7. Optimize $\boldsymbol{t}_{ij}$ with Eq. (29)
Ensure: The retargeted matrix $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2 \cdots \boldsymbol{t}_n]^{\mathrm{T}} \in \mathbb{R}^{N \times C}$.
```

(27) can be further reduced to a series of subproblems, and the jth one can be written as

$$\min_{\boldsymbol{t}_{ij}} \left(\boldsymbol{p}_{ij} - \boldsymbol{t}_{ij}\right)^2, \quad \text{s.t.} \quad \boldsymbol{p}_{ik} + \boldsymbol{\rho} - \boldsymbol{t}_{ij} \geq c \quad \forall j \neq k. \tag{28}$$

Define an auxiliary term $\boldsymbol{\kappa} \in \mathbb{R}^C$, whose jth element should be $\boldsymbol{\kappa}_j = \boldsymbol{t}_{ij} + c - \boldsymbol{t}_{ik}$, where $\boldsymbol{\kappa}_j \leq 0$ represents the optimal value, otherwise a unsatisfactory target. From the basic knowledge of quadratic programming, the solution to problem (28) can be obtained as follows:

$$\boldsymbol{t}_{ij} = \begin{cases} \boldsymbol{p}_{ij} + \boldsymbol{\rho}, & \text{if} \quad j = k \\ \boldsymbol{p}_{ij} + \min\left(\boldsymbol{\rho} - \boldsymbol{\kappa}_j, 0\right), & \text{otherwise}. \end{cases} \tag{29}$$

In iteration process, the parameter $\boldsymbol{\rho}$ in (29) should be updated to achieve the optimal $\boldsymbol{t}_{ij}$. To do this, we substitute (29) into problem (27), and obtain an optimization problem with respect to $\boldsymbol{\rho}$ as follows:

$$\arg\min_{\boldsymbol{\rho}} \boldsymbol{\rho}^2 + \sum_{j \neq k} \left(\min\left(\boldsymbol{\rho} - \boldsymbol{\kappa}_j, 0\right)\right)^2. \tag{30}$$

Obviously, here we just set the first-order derivation of (30) with respect to $\boldsymbol{\rho}$ to be 0, and the optimal value of $\boldsymbol{\rho}$ can be obtained as follows:

$$\boldsymbol{\rho} = \frac{\sum_{j \neq k} \boldsymbol{\kappa}_j \Upsilon\left(\Phi'\left(\boldsymbol{\kappa}_j\right) > 0\right)}{1 + \sum_{j \neq k} \boldsymbol{\kappa}_j \Upsilon\left(\Phi'\left(\boldsymbol{\kappa}_j\right) > 0\right)} \tag{31}$$

where $\Upsilon(\cdot)$ denotes an indicator operator, i.e., $\Upsilon(\cdot) = 1$ when the conditions are met, otherwise $\Upsilon(\cdot) = 0$. Algorithm 2 presents the specific process to optimize the ith row of $\boldsymbol{T}$, where the constant c is set to 1. According to the obtained optimal $\boldsymbol{T}$ and similar optimization procedure of LDMBLS, we can get the optimal solution of $\boldsymbol{W}$ as follows:

$$\boldsymbol{W} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_1 \boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A} + \lambda_2 \boldsymbol{I}\right)^{-1} \boldsymbol{A}^{\mathrm{T}}\boldsymbol{T}. \tag{32}$$

With the optimization, the sequence $\{\boldsymbol{W}^{(t)}, \boldsymbol{T}^{(t)}\}$ will be stable at a certain point, and then the iteration can be terminated. The complete processes to solve (17) are summarized in Algorithm 3.

Algorithm 3 Optimizing (17) by ADMM

```
Require: Broad transformed feature $\boldsymbol{A}$; binary label matrix $\boldsymbol{Y}$; the regularization parameters $\lambda_1$ and $\lambda_2$;
    1. Initialize $\boldsymbol{W} = \boldsymbol{0}$;
    2. Calculate the Laplacian matrix $\boldsymbol{L}$ with Eq. (9);
    Set iteration number $t = 1$;
    while not converged do
        3. Optimize the retargeted matrix $\boldsymbol{T}$ by Algorithm 2;
        4. Optimize $\boldsymbol{W}$ with Eq. (32);
        5. Update iteration number $t = t + 1$;
    end while
Ensure: Output weight matrix $\boldsymbol{W}$.
```

C. Computational Complexity and Convergence

In this part, we will discuss the computational complexity and convergence of our proposed iterative procedure to solve (16) and (17). For ease of expression, we denote the broad transformed feature $\boldsymbol{A} \in \mathbb{R}^{N \times d}$, where N and d, respectively, represents the number and dimension of $\boldsymbol{A}$.

Our starting point is the analysis of the computational complexity. From Algorithm 1, we can see the iterative process and constructing the Laplacican matrix are the main computational burden, and the time-consuming in simple matrix multiplication and addition can be ignored. About constructing $\boldsymbol{L}$, we know the complexity depends on the calculation of the adjacency matrix $\boldsymbol{V}$, i.e., $\mathcal{O}(N^2 d)$. For the optimization of $\boldsymbol{W}$ in one iteration step, we can see there exists a large proportion of the inverse operation $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_1 \boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{A} + \lambda_2 \boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}$, whose computational complexity can be accumulated as $\mathcal{O}(d^3 + d^2 N)$. Thus, the total computational time complexity in optimizing (16) is expressed as $\mathcal{O}(t(d^3 + d^2 N) + N^2 d)$, where t is the iteration number. Similarly, for the REM-BLS model, the only modification states the construction of retargeted matrix $\boldsymbol{T}$. From Algorithm 2, we can see the complexity of this part is $\mathcal{O}(NC)$. Therefore, the overall computational time complexity in Algorithm 3 can be expressed as $\mathcal{O}(t(d^3 + d^2 N + NC) + N^2 d)$, where t also means the number of iterations. It is obvious that the complexity of these two proposed algorithms is on the same order of magnitude.

Now, let us analyze the convergence of Algorithms 1 and 3, respectively. For the iteration process in Algorithm 1, the objective function values can decrease to a minimum point with the computation block $\{\boldsymbol{W}, \boldsymbol{M}\}$ recursively repeated, which can be guaranteed by the following theorem.

Theorem 1: Denote (16) as $F(\boldsymbol{W}, \boldsymbol{M})$. If the optimal results of the tth iteration are $\{\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)}\}$, the sequence $F(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)})$ can converge to a minimum point within the iterations.

Proof: First, given $\boldsymbol{M}^{(t-1)}$, the optimal $\boldsymbol{W}^{(t)}$ can be calculated by minimizing the following function:

$$\arg\min_{\boldsymbol{W}} F\left(\boldsymbol{W}, \boldsymbol{M}^{(t-1)}\right) \tag{33}$$

which is a convex optimization problem. As the optimization process is to minimize the objective function, we can get

$$F\left(\boldsymbol{W}^{(t-1)}, \boldsymbol{M}^{(t-1)}\right) \geq F\left(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t-1)}\right). \tag{34}$$

Next, (25) is solved by fixing $\boldsymbol{W}^{(t)}$ to get $\boldsymbol{M}^{(t)}$. We can also find that the subproblem with respect to $\boldsymbol{M}$ is convex, then we can get

$$F\left(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t-1)}\right) \geq F\left(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)}\right). \tag{35}$$

Combining (34) and (35), we can get

$$F\left(\boldsymbol{W}^{(t-1)}, \boldsymbol{M}^{(t-1)}\right) \geq F\left(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)}\right). \tag{36}$$

Thus, the successive $F(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)})$ should be a monotonically decreasing sequence with the iteration process.

Second, since all three parts composing (16) are nonnegative, the objective function $F(\boldsymbol{W}, \boldsymbol{M})$ should also be nonnegative. This implies that the sequence $F(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)})$ can have a lower-bound of 0. Then, according to the Monotone Convergence Theorem in [23], we can conclude that the sequence $F(\boldsymbol{W}^{(t)}, \boldsymbol{M}^{(t)})$ should be convergent. Moreover, considering (16) is a joint convex model, any local minimum can be a global minimum [16], so the objective function value can converge to a global minimum point within the iterations. Therefore, the above theorem is successfully proved. ■

For Algorithm 3, its convergence can be guaranteed by the following theorem.

Theorem 2: Denote (17) as $\Gamma(\boldsymbol{W}, \boldsymbol{T})$. If the optimal results of the tth iteration are $\{\boldsymbol{W}^{(t)}, \boldsymbol{T}^{(t)}\}$, the sequence $\Gamma(\boldsymbol{W}^{(t)}, \boldsymbol{T}^{(t)})$ can converge to a minimum point within the iterations

Proof: Since the optimization process of REMBLS is similar to that of LDMBLS, i.e., it is still solved by variable fixed iteratively, the proof of Theorem 2 follows a similar structure to that of Theorem 1. We do not go into details here. ■

In summary, our proposed flexible label-induced BLS models aim to provide more flexibility in the label assignment process, aligning it better with the manifold assumption and improving the overall performance of the proposed framework. The complete details are demonstrated in Algorithm 4.

```
Algorithm 4 Flexible Label-Induced BLS Models
Require: Training sample matrix $X$; label matrix $Y$; feature mapping function $\phi(\cdot)$; activation function $\xi(\cdot)$; network parameters $N_g$, $N_f$, and $N_e$; Laplacian matrix $L$; regularization parameters $\lambda_1$ and $\lambda_2$.
    1. Random $W_{e_i}, \beta_{e_i}, i = 1, 2, \ldots, N_g$;
    2. Calculate $Z_i = \phi_i(X W_{e_i} + \beta_{e_i}), i = 1, 2, \ldots, N_g$, combining the mapping features generated by the feature nodes $Z^{N_g} \triangleq [Z_1, Z_2, \ldots, Z_{N_g}]$;
    3. Random $W_{h_j}, \beta_{h_j}, j = 1, 2, \ldots, m$;
    4. Calculate $H_j = \xi_j(Z^{N_g} W_{h_j} + \beta_{h_j}), j = 1, 2, \ldots, m$, generating enhancement features by Combining mapping features $H^m \triangleq [H_1, H_2, \ldots, H_m]$;
    5. Set the broad-transformed as $A \triangleq [Z^{N_g}, H^m]$;
    6. Calculate the projection matrix $W$ by Algorithm 1 (LDMBLS) or Algorithm 3 (REMBLS);
Ensure: The projection matrix $W$.
```

V. Experiments

In this section, extensive experiments on several popular multiclass recognition tasks are conducted to verify the effectiveness of our proposed LDMBLS and REMBLS. Table I describes the general statistics of the seven used databases. We select some state-of-the-art algorithms as competitors, including ELM [24], SRC [1], CRC [25], SVM [26], DLSR [15], GBLS [27], ReLSR [16], IDLSR [28], RLR [29], standard BLS [7], SGLDBLS [30], DRAGD [31], Fisher [32], DRC [33], and MDBLS [34]. The reported results of these comparison methods are obtained either from the relevant papers or by rerunning the source codes provided by the authors. For ELM, we set the sigmoid function $g(x) = 1/(1 + \exp(-x))$ as activation function, while the $\tanh(x) = \sinh(x)/\cosh(x)$ is generally utilized in the BLS-based methods (BLS, GBLS, SGLDBLS, MDBLS, LDMBLS, REMBLS). For the methods that incorporate graph regularization (RLR, GBLS, LDMBLS, REMBLS), the heat kernel parameter σ is set to 1. The optimal regularization parameters for all the methods are determined through a grid search strategy over a wide range set $\{10^{-8}, 10^{-7}, \ldots, 10^{-1}\}$ to ensure the best performance of each algorithm. The specific setting of network parameters (N_g, N_f, N_e) in LDMBLS and REMBLS should be provided in the subsequent recognition tasks. All experiments in this section are implemented in MATLAB R2016b, using a 3.10-GHz CPU and 8-GB memory.

TABLE I
Brief Descriptions of the Databases Used in Experiments

Databases	# Classes	# Features	# Samples
Extended YaleB	38	1,024	2,414
AR	120	2,000	3,120
LFW	86	1,024	1,251
Fifteen Scene	15	3,000	4,485
COIL20	20	1,024	1,440
MNIST	10	784	70,000
USPS	10	256	9,298

A. Evaluations on Face Databases

First, the recognition capability of our proposed methods is evaluated on three popular face databases: Extended YaleB [35], AR [36], and LFW [37], whose detailed introduction is as follows.

1) *Extended YaleB [35]:* This dataset contains 2414 front face images of 38 individuals, and each individual has around 64 near frontal images with different expressions and outfits. Each image in this database for our experiments has been resized to 32 × 32 pixels in advance. We randomly select {10, 15, 20, 25} images per subject for training, and the rest is used for testing.
2) *AR [36]:* The original AR dataset contains about 4000 color face images of 126 subjects. In this part, a popular subset (i.e., 3120 samples of 120 subjects) is selected for experiments, and all these image samples have been resized to 50 × 40 pixels in advance. During the experiments, {4, 6, 8, 12} images per subject are randomly chosen for training and the remaining for testing.
3) *LFW [37]:* There are more than 13 000 face images in its original version, and about 1680 people contain more

TABLE II

RECOGNITION RESULTS (AVER ± STD%) ON THE EXTENDED YALEB DATABASE

Method	10 train	15 train	20 train	25 train
SVM	87.35 ± 1.22	89.64 ± 0.98	93.65 ± 0.75	96.12 ± 0.59
ELM	82.98 ± 1.22	88.94 ± 0.98	93.17 ± 0.85	95.97 ± 0.72
CRC	88.76 ± 1.21	92.77 ± 1.08	94.97 ± 0.96	96.04 ± 0.87
SRC	85.37 ± 0.49	91.17 ± 0.53	94.48 ± 0.71	96.54 ± 0.54
DRC	90.22 ± 0.93	93.17 ± 0.84	96.64 ± 0.85	97.95 ± 0.96
DLSR	85.29 ± 0.73	91.14 ± 0.76	93.85 ± 0.53	95.71 ± 0.61
IDLSR	89.44 ± 0.93	93.89 ± 0.79	96.80 ± 1.04	97.62 ± 0.96
ReLSR	87.53 ± 0.93	92.98 ± 0.89	95.36 ± 0.77	96.20 ± 0.85
RLR	88.53 ± 1.29	92.83 ± 0.90	95.74 ± 0.66	96.90 ± 0.35
BLS	86.53 ± 0.62	92.06 ± 0.93	95.64 ± 0.69	96.45 ± 0.74
GBLS	88.45 ± 1.29	93.59 ± 0.95	95.47 ± 0.58	97.04 ± 0.55
SGLDBLS	91.06 ± 0.91	95.33 ± 0.85	97.47 ± 0.75	98.82 ± 0.86
DRAGD	**91.14 ± 0.98**	95.09 ± 1.11	96.95 ± 0.48	98.05 ± 0.48
Fisher	90.30 ± 0.91	94.53 ± 0.52	96.37 ± 0.62	97.75 ± 0.19
MDBLS	90.41 ± 0.74	95.12 ± 0.58	97.46 ± 0.48	98.57 ± 0.47
LDMBLS	90.31 ± 1.13	**95.43 ± 0.61**	**97.70 ± 0.76**	98.57 ± 0.41
REMBLS	90.46 ± 0.91	95.16 ± 0.98	97.52 ± 0.53	**98.88 ± 0.77**

TABLE III

RECOGNITION RESULTS (AVER ± STD%) ON THE AR DATABASE

Method	4 train	6 train	8 train	12 train
SVM	68.76 ± 1.58	82.98 ± 0.94	89.91 ± 0.83	95.61 ± 0.94
ELM	85.17 ± 1.03	91.71 ± 0.92	94.12 ± 0.96	97.73 ± 1.12
CRC	88.66 ± 1.03	93.18 ± 0.86	95.02 ± 0.83	96.91 ± 0.87
SRC	84.39 ± 0.99	89.63 ± 0.76	93.49 ± 1.06	95.30 ± 0.79
DRC	90.74 ± 0.74	94.92 ± 0.69	96.30 ± 0.76	98.96 ± 0.68
DLSR	86.34 ± 0.66	91.35 ± 0.74	94.78 ± 0.83	97.27 ± 0.59
IDLSR	91.63 ± 0.77	95.81 ± 0.64	97.39 ± 0.83	98.75 ± 0.62
ReLSR	88.34 ± 0.85	93.03 ± 0.74	95.62 ± 0.92	97.57 ± 1.03
RLR	91.38 ± 0.51	95.36 ± 0.76	97.17 ± 0.36	98.04 ± 0.25
BLS	86.36 ± 0.61	92.13 ± 0.72	95.14 ± 0.55	98.04 ± 0.62
GBLS	89.98 ± 1.40	92.74 ± 0.53	95.89 ± 0.61	97.93 ± 0.26
SGLDBLS	92.65 ± 0.83	**96.59 ± 0.42**	97.81 ± 0.67	99.25 ± 0.74
DRAGD	93.00 ± 0.55	96.28 ± 0.66	97.12 ± 0.44	98.89 ± 0.32
Fisher	86.03 ± 1.14	95.32 ± 0.65	97.25 ± 0.45	97.34 ± 0.41
MDBLS	92.01 ± 0.63	96.00 ± 0.72	97.41 ± 0.61	99.23 ± 0.59
LDMBLS	92.01 ± 0.26	95.88 ± 0.79	**97.92 ± 0.84**	99.17 ± 0.52
REMBLS	**93.11 ± 1.15**	95.83 ± 0.52	97.50 ± 0.67	**99.31 ± 0.48**

TABLE IV

RECOGNITION RESULTS (AVER ± STD%) ON THE LFW DATABASE

Method	5 train	6 train	7 train	8 train
SVM	26.04 ± 1.58	29.52 ± 1.63	30.60 ± 1.39	33.14 ± 1.48
ELM	35.08 ± 1.59	38.37 ± 1.67	40.68 ± 0.96	42.45 ± 1.47
CRC	30.12 ± 1.14	31.44 ± 1.18	32.51 ± 1.26	34.55 ± 1.29
SRC	29.38 ± 1.52	32.51 ± 1.49	33.64 ± 1.68	35.12 ± 1.93
DRC	35.08 ± 1.23	40.14 ± 1.35	43.68 ± 1.39	44.82 ± 1.57
DLSR	27.90 ± 0.87	30.80 ± 1.45	33.73 ± 1.52	36.80 ± 1.49
IDLSR	37.64 ± 1.17	40.63 ± 1.26	42.58 ± 1.19	44.47 ± 1.32
ReLSR	31.81 ± 1.53	34.45 ± 0.94	37.70 ± 1.48	40.37 ± 1.39
RLR	35.81 ± 1.11	37.82 ± 0.95	39.85 ± 1.47	40.04 ± 1.51
BLS	28.38 ± 1.42	31.56 ± 1.36	35.26 ± 1.52	37.30 ± 1.79
GBLS	35.20 ± 0.74	38.48 ± 1.51	38.89 ± 1.17	42.70 ± 1.23
SGLDBLS	38.36 ± 1.12	41.81 ± 1.36	44.06 ± 1.97	45.64 ± 1.19
DRAGD	38.78 ± 1.23	42.78 ± 1.38	45.02 ± 1.11	46.61 ± 1.59
Fisher	37.02 ± 1.41	40.97 ± 1.35	43.77 ± 1.40	45.25 ± 1.05
MDBLS	37.64 ± 1.15	41.36 ± 1.32	43.91 ± 1.27	45.29 ± 1.33
LDMBLS	38.49 ± 1.06	41.36 ± 1.02	44.07 ± 1.71	45.33 ± 0.93
REMBLS	**39.10 ± 1.71**	**42.88 ± 1.52**	**45.30 ± 1.53**	**47.96 ± 2.23**

than two face images. In this experiment, 1251 images from 86 individuals are utilized, in which each subject can provide about 11 images. All these images are uniformly resized to 32 × 32 pixels for experiments. We also randomly choose $\{5, 6, 7, 8\}$ samples in each class for training and the rest for testing.

Following the intended experimental setup, we repeat each experiment for ten times to eliminate the random disturbance and make the experimental results convincing. The grid search method is operated for determining the optimal network parameters (N_g, N_f, N_e) on $[1, 2, \ldots, 10] \times [10, 11, \ldots, 20] \times [500, 510, \ldots, 1000]$. Then the average recognition accuracy with the standard derivation (Aver ± Std) is reported as the final comparison indicator. Tables II–IV show the classification results of our proposed methods as well as those competitors on the Extended YaleB, AR, and LFW dataset, respectively, in which the bold number indicates the best performance. From Tables II–IV, we can see our proposed two models can perform better in comparison with other algorithms. Specifically, for the Extended YaleB dataset, LDMBLS and REMBLS can obtain the best recognition effects when the training number is 15, 20, and 25, respectively. And the difference between these two methods is only about 0.2%–0.3%. For the AR database, though SGLDBLS achieves the best accuracy with 6 train, we can see our proposed two methods can still perform the best in other cases with significant advantages. Since the images contained in LFW are intrinsically difficult to be identified, the accuracies obtained by different classification algorithms are relatively low. However, we can see our proposed LDMBLS method can reach 38.49%, 41.36%, 44.07%, and 45.33% with $\{5, 6, 7, 8\}$ train. Moreover, REMBLS can further improve the accuracies to 39.10%, 42.88%, 45.30%, and 47.96%, which has obviously surpassed other comparison methods. Overall, we can find that incorporating the flexible label strategy into the framework of manifold BLSs can indeed be helpful for improving the classification results on the multiclass face recognition problem.

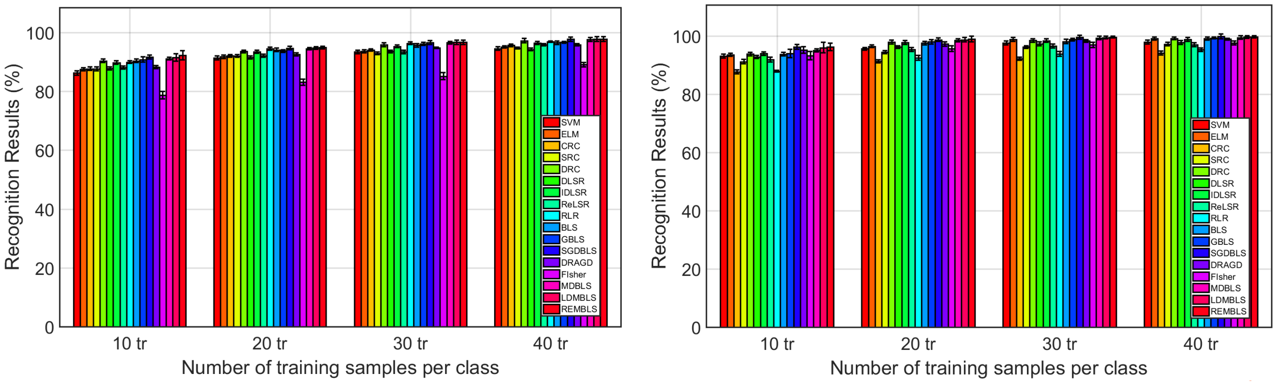
B. Evaluations on Scene and Object Databases

In this section, we continue to test the effectiveness of our proposed LDMBLS and REMBLS in the scene and object recognition fields. Here the popular Fifteen Scene Categories [38] and COIL20 [39] databases are selected as benchmarks, whose detailed introductions are as follows.

1) *Fifteen Scenes [38]:* This database contains 15 kinds of scenarios, such as kitchen, store, bedroom,

(a) (b)

Fig. 4. Partial images from (a) Fifteen Scenes Categories database and (b) COIL20 database.



(a) (b)

Fig. 5. Recognition results (Aver ± Std%) of different methods on (a) Fifteen Scene Categories database and (b) COIL20 database.

MITforest, MITcoast, with a total number of 4485 images. Due to the complexity of the original pictures, here we extract the spatial pyramid features with dimensions = 3000 for experimentation. We randomly select {10, 20, 30, 40} samples per category for training and the rest for testing.

2) *COIL20 [39]:* As its name indicates, this database collects 1440 gray image samples from 20 objects. For each object, images are acquired every 5°, then a total of 72 pictures can be reached. We resize each image to be 32 × 32 pixels, then select {10, 15, 20, 25} samples per class for training and the remaining for testing.

To provide readers with a more intuitive understanding of these two databases, we select several sample images which are exhibited in Fig. 4. Then the comparison experiments can be conducted according to the predefined scheme. Each subexperiment is repeated for ten times by combining the training and test samples randomly. And we still treat the average recognition accuracy with the standard derivation (Aver ± Std) as the comparison metric. To get the optimal (N_g, N_f, N_e), the grid search is utilized on $[1, 2, \ldots, 10] \times [10, 12, \ldots, 40] \times [400, 450, \ldots, 1600]$ in this part. The recognition results of our proposed algorithms and other competitors on these two databases are depicted in Fig. 5, in which each histogram represents the classification accuracy of certain method with a training number. We can see the performances of all algorithms can be improved as the number of training samples increases. Compared with other competitive algorithms, it is evident that both our proposed LDMBLS and REMBLS method can always achieve better recognition accuracies under different numbers of training samples. Furthermore, we can also find that the performance of REMBLS which imposes the retargeted relaxation regression objective achieves the best property in nearly all the cases. This phenomenon not only shows that relaxing the margins of different classes can be more beneficial for the regression but also reflects the necessity to relax the targets by jointly learning the labels with the output weights in the BLS method. All in all, the experimental results further illustrate the potential and application prospects of these two proposed methods for the object and scene recognition problems.

C. Evaluations on Handwritten Recognition

In this section, we compare our proposed LDMBLS and REMBLS to verify the characteristics of two different slack label strategies. The standard BLS is chosen as baseline for comparison, and two popular handwritten databases: MNIST [40] and USPS [41], are utilized as benchmarks.

First, we compare their performances in classification accuracies and optimal network parameters setting with different number of training samples. For MNIST, we split the 60 000 training data into a validation set with 1000 images per subject and randomly select {100, 200, 400, 600, 1000} images per subject from the rest for training, the whole test set with 10 000 sample for testing. The parameters (N_g, N_f, N_e) are determined on $[5, 6, \ldots, 40] \times [12, 11, \ldots, 45] \times [600, 650, \ldots, 1800]$ by grid search. For USPS, we randomly choose 300 images per class for validation and further select {100, 150, 200, 250} images per class for training, the remaining images for testing. The network parameters can be searched on the range $[5, 6, \ldots, 35] \times [10, 11, \ldots, 45] \times [200, 210, \ldots, 1000]$. The generated validation set is used for model selection, and the random generation process is repeated for ten runs to avoid interference. Tables V and VI show the comparison results, in which Acc$-\mathcal{V}$ and Acc$-\mathcal{T}$ denotes the recognition results on validation and test set, respectively. We can see the optimal network parameters of these three methods are nearly on the same order of magnitude,

TABLE V

COMPARISON RESULTS (AVER ± STD%) BETWEEN BLS, LDMBLS, AND REMBLS ON MNIST

Number	BLS					LDMBLS					REMBLS				
	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)
100 tr	7	18	600	90.91±0.50	91.67±0.27	8	14	600	93.02±0.33	93.15±0.20	8	15	600	**93.29±0.41**	**93.40±0.26**
200 tr	19	20	750	94.44±0.09	94.55±0.07	10	15	700	94.73±0.24	94.78±0.22	14	20	900	**94.86±0.21**	**94.87±0.04**
400 tr	15	30	1000	95.42±0.05	95.74±0.15	18	25	1000	95.94±0.14	96.08±0.06	20	35	1300	**96.15±0.12**	**96.20±0.07**
600 tr	25	35	1200	96.18±0.22	96.22±0.09	25	30	1000	96.37±0.15	96.55±0.10	25	40	1300	**96.61±0.19**	**96.65±0.07**
1000 tr	30	45	1500	96.84±0.08	96.98±0.04	35	40	1200	96.98±0.14	97.22±0.12	40	45	1700	**97.14±0.15**	**97.26±0.06**

TABLE VI

COMPARISON RESULTS (AVER ± STD%) BETWEEN BLS, LDMBLS, AND REMBLS ON USPS

Number	BLS					LDMBLS					REMBLS				
	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)	N_g	N_f	N_e	Acc$-\mathcal{V}$(%)	Acc$-\mathcal{T}$(%)
100 tr	7	15	200	88.90±0.09	91.17±0.48	5	10	150	94.15±0.45	94.84±0.25	7	14	300	**95.10±0.59**	**95.59±0.27**
150 tr	10	20	250	92.70±0.58	93.41±0.29	15	20	300	95.10±0.55	95.50±0.20	20	20	400	**95.30±0.26**	**96.19±0.21**
200 tr	25	35	500	93.95±0.68	94.34±0.04	18	30	600	95.85±0.91	96.38±0.18	30	30	730	**96.65±0.33**	**96.70±0.10**
250 tr	30	40	1000	94.80±0.38	94.98±0.19	20	35	800	96.35±0.50	96.64±0.27	35	40	900	**96.45±0.43**	**97.29±0.31**

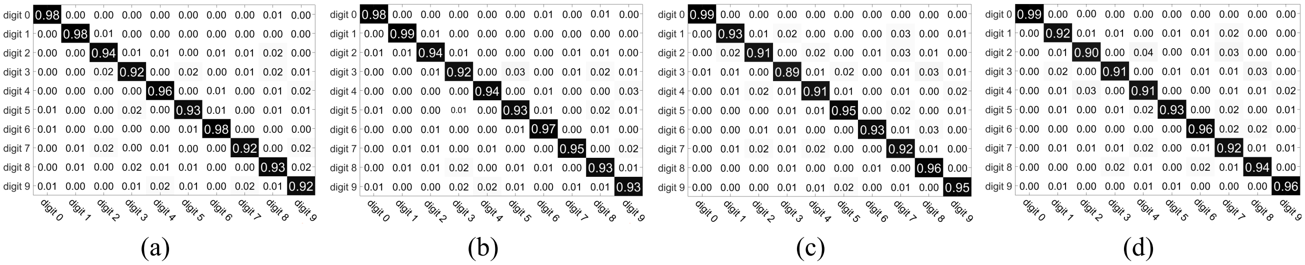


Fig. 6. Confusion matrix of (a) LDMBLS and (b) REMBLS on MNIST with 200 tr, and (c) LDMBLS and (d) REMBLS on USPS with 100 tr.

which reflects that manifold learning and flexible label strategy do not affect the methods' operational efficiency. Obviously, LDMBLS performs better than BLS, and REMBLS can further improve the recognition accuracy in all cases. And this improvement is more significant in the small sample scenario. These phenomena can verify the effectiveness and robust of the proposed retargeted strategy well.

Second, we characterize the confusion matrix to display the class-specific recognition abilities of our proposed methods. Generally, the diagonal entries of a confusion matrix can represent the recognition accuracy on each class, the larger the better. Here, the MNIST with 200 tr and USPS with 100 tr per class are utilized, and the obtained results are visualized in Fig. 6. We can find the diagonal elements of these four matrices are absolutely dominant and the whole off-diagonal elements are microscopic. This reflects that our proposed methods both can identify the specific class data well. REMBLS can still perform better than LDMBLS in most cases. Nevertheless, we should point out that our algorithms still have limitations in recognizing certain digits, like 3, 5, and 8. The reason states that the similarities between these digits are a bit high, which brings difficulty to the samples' correct distinction. And how to solve this deficiency should be one of our future work.

Furthermore, we measure the efficiency of the proposed two slack label strategies with more classification metrics, including kappa coefficient, micro-precision, macro-precision, micro-F1, macro-F1, micro-recall, macro-recall, and accuracy. The MNIST with {400, 600} tr and USPS with {150, 250} tr per class are chosen for experiments. The obtained results are shown with four radar charts in Fig. 7. We can find our proposed methods can achieve excellent performance on nearly all these metrics. Though the label dragging manner cannot perform well on the kappa coefficient on MNIST, the retargeted strategy can still attain a satisfactory effect in this scenario. Considering the performances of our proposed methods on all these metrics, it is convincing that our proposed flexible label strategies, especially the retargeted method, can indeed be beneficial for multiclass recognition.

D. Experimental Analysis

In this section, several further conclusions drawn from the experimental results should be analyzed as follows.

1) Under the supervision of label information, linear regression-based methods, namely DLSR, ReLSR, and IDLSR, generally outperform representation-based methods like SRC and CRC. And the recognition results can be further improved by IDLSR which equips with the structural consistency within classes. A similar trend appears in the comparison between DRC and other representation-based methods. Hence, the inclusion of

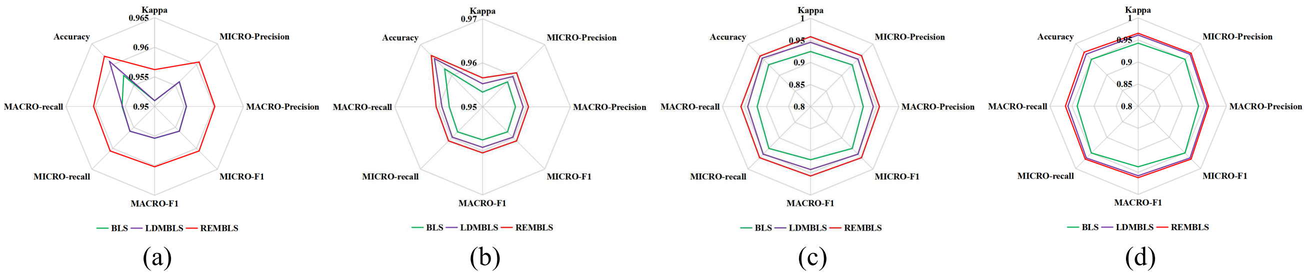


Fig. 7. Radar chart of different methods on (a) MNIST with 400 tr, (b) MNIST with 600 tr, (c) USPS with 150 tr, and (d) USPS with 250 tr.

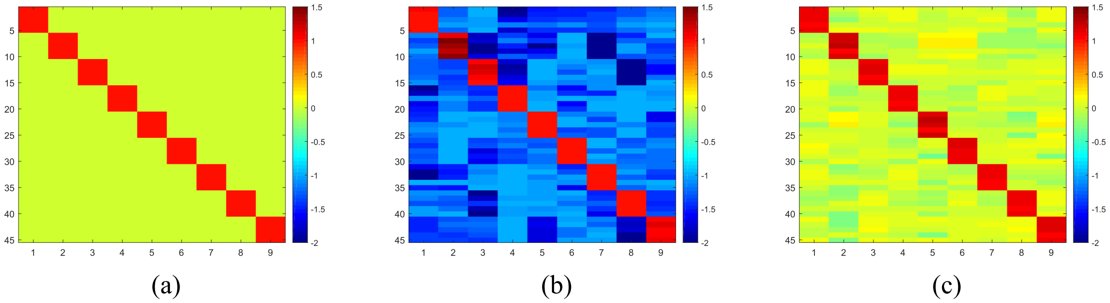


Fig. 8. Visualization of (a) strict binary label, (b) dragging label, and (c) retargeted label on the Extended YaleB database.

class-specific sparse information should be considered as crucial prior knowledge in modeling.

2) In most cases, BLS-based methods can achieve higher recognition results compared to other classifiers, such as SVM and ELM. The incorporation of graph regularization enhances the performance of GBLS beyond the standard BLS method. Then, by jointly incorporating the slack labels and feature selection into the BLS framework, the output weights in SGLDBLS can be more discriminative which shows great superiority for multiclass recognition. However, as the locally invariant property of data has not been fully considered, the overfitting problem is still relatively severe, which also leaves room for further improvement in the recognition problem of the algorithm.

3) Apparently, our proposed two BLS-based methods: LDMBLS and REMBLS, can be more discriminative than all the other competitive classifiers for each recognition task. The reasons are as follows: a) both of these two methods are based on the broad network architecture which can produce rich and representative features for recognition and b) jointly with the manifold regularization, the introduced flexible labels in LDMBLS and REMBLS can make the margins between different samples more appropriate, which is beneficial for learning compact weights and eliminating the overfitting problem.

4) Compared with label dragging in LDMBLS, it can be found that direct learning the regression targets simultaneously with the output weights in REMBLS can always achieve higher performances for multiclass recognition. Such a scenario just confirms the strong supervisory role of the new flexible labels in the latent manifold space, and the locally invariant property of data can be better preserved. Consequently, all evaluations strongly confirm the effectiveness of REMBLS and the necessity of embedding flexible label information with manifold technology in the modeling process of the broad network.

E. Property of the Flexible Labels

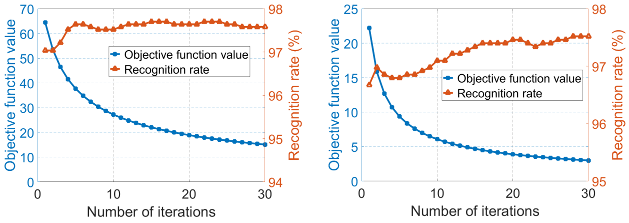
To guide the strict binary label of BLS into a soft variable matrix, we proposed two flexible strategies with a manifold manner. To highlight this distinguishable property, we first visualize the obtained flexible labels alongside the original strict binary labels on Extended YaleB with 20 tr per class in Fig. 8. The highlighted parts in the heatmap correspond to large entries of the label matrix. The two flexible label matrix both exhibit a similar block diagonal structure as the strict binary labels, and this correspondence can retain the valid supervision information in the original labels. Meanwhile, the off-diagonal elements of the flexible labels are noticeably nonzero, which reflects the hard targets are relieved in a sense. Thus, the obtained labels can be more distinguishable and provide more freedom for the approximation process. Moreover, we randomly select a portion of data and calculate the marginal values of their soft labels by two proposed methods. The results are shown in Table VII, in which ci-j means the jth sample from the ith class. We can see the margins between different classes are indeed enlarged. Comparatively, the margin adjustment by REMBLS can be milder, which can also be reflected by the heatmap in Fig. 8(c). This phenomenon just can explain why REMBLS can perform better than LDMBLS in most cases, i.e., the interclass separability can be enhanced and simultaneously ensure the intraclass compactness. Overall, these flexible labels can offer substantial assurance for the accurate recognition.

F. Convergence Analysis

There are iteration processes in optimizing our proposed LDMBLS and REMBLS, their convergence properties can

TABLE VII
MARGINS OF THE SOFT LABELS BETWEEN DIFFERENT SAMPLES

	LDMBLS				REMBLS			
	C1-1	C1-2	C2-1	C2-2	C1-1	C1-2	C2-1	C2-2
C1-1	0.00	0.714	4.076	4.235	0.00	0.144	1.721	1.758
C1-2	–	0.00	4.427	4.573	–	0.00	1.826	1.866
C2-1	–	–	0.00	0.301	–	–	0.00	0.131
C2-2	–	–	–	0.00	–	–	–	0.00



(a)

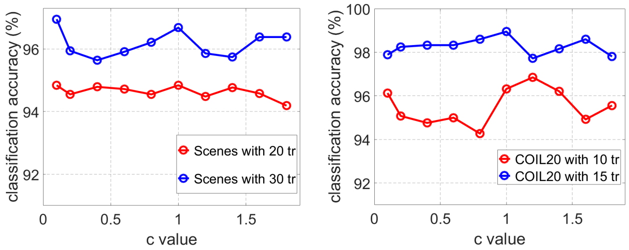
(b)

Fig. 9. Curves of the objective function values and recognition rates versus the number of iterations for (a) LDMBLS and (b) REMBLS.

have a significant impact on their efficiency and recognition performances. Such a convergence behavior has been guaranteed in theory in Section IV-C, and here we should further demonstrate it from an experimental perspective. To do this, not only the objective function values but also the recognition rates of each method on the Extended YaleB with 20 tr per class versus the number of iterations are visualized in Fig. 9. The iteration number is set as 30. It is evident that the recognition accuracy improves significantly in the initial iterations and then tends to stabilize. Similarly, as the number of iterations increases, the objective values initially decrease and then stabilize. Thus, the good convergence performances of our proposed methods can be proved well. What is more, both the stabilization points of the objective value and recognition rate can be achieved within 30 iterations. In fact, in certain experiments, an even smaller number of iterations may suffice. Considering the low complexity in each iteration, the efficiency of our proposed methods can be well guaranteed.

G. Parameters Selection

In REMBLS, we introduced a marginalized constraint c to ensure flexibility and separability in the relearned labels between true and false categories. For ease of implementation, it is set to 1 empirically. Here, we should provide a discussion on the effect of different c values on REMBLS. The Fifteen Scenes with $\{20, 30\}$ tr per class and COIL20 with $\{10, 15\}$ tr per class are utilized. The experimental results on these two databases are shown in Fig. 10. We can see the recognition rate exhibits variation with different c values. Nevertheless, considering the experimental results in Section V-B, it is evident that the effectiveness of REMBLS can be maintained under different c values. Hence, the effect of the c value on the model is relatively weak and controllable. Furthermore, we observed that the best recognition result is achieved with different c values on different databases. Therefore, setting c to 1 is not the only option. In certain specific applications, REMBLS can perform better if we carefully determine the most appropriate c value.



(a)

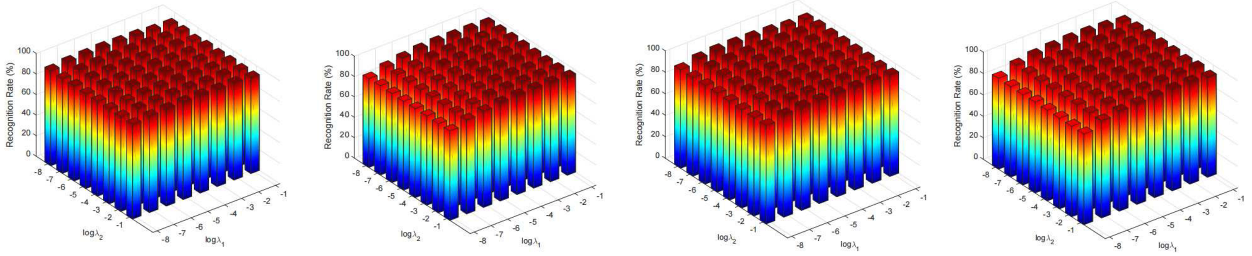
(b)

Fig. 10. Recognition rates of REMBLS with different c values on (a) Fifteen Scenes and (b) COIL20.

TABLE VIII
ABLATION RESULTS OF LDMBLS AND REMBLS ON THE FIFTEEN SCENES AND COIL20 DATABASE

Method	Fifteen Scenes		COIL20	
	20 train	40 train	15 train	25 train
$\lambda_2 = 0$	94.27±0.57	97.07±0.25	98.25±1.70	99.57±0.43
$\lambda_1 = 0$	92.11±1.16	96.45±0.33	97.57±0.59	99.36±0.34
LDMBLS	94.72±0.45	97.71±0.83	98.77±0.75	99.68±0.47
$\lambda_2 = 0$	94.60±0.62	97.30±0.34	98.51±0.70	99.68±0.16
$\lambda_1 = 0$	93.26±0.45	96.94±0.47	97.72±0.84	99.47±0.12
REMBLS	94.84±0.47	97.73±0.88	98.95±0.97	99.70±0.39

Then, we analyze the selection of the regularization parameters λ_1 and λ_2 in our proposed LDMBLS and REMBLS. These two parameters govern the trade-off between the manifold regularizer and the Frobenius norm of $\boldsymbol{W}$, which can have an influence on the final recognition rate. To discover the relationship between their selections and the recognition rates, we test the performances of our proposed methods on AR (8 tr per class) and Extended YaleB (20 tr per class) with a range of regularization parameters. Empirically, the candidate sets of λ_1, λ_2 in LDMBLS and REMBLS are both set as $\{10^{-8}, 10^{-7}, \ldots, 10^{-1}\}$. The recognition results with the variations of λ_1, λ_2 are illustrated in Fig. 11. We can observe that the performances of LDMBLS and REMBLS are not very sensitive to the settings of λ_1 and λ_2. In general, the recognition accuracies of our proposed methods cannot be seriously disturbed when these two parameters are not so large. Nevertheless, it is worth noting that both these two constraint terms are indispensable in our proposed methods. To verify this, we further design ablation experiments on the Fifteen Scenes and COIL20 database, in which LDMBLS and REMBLS are, respectively, compared with their variants: $\lambda_2 = 0$ means "only-manifold" and $\lambda_1 = 0$ means "only-Frobenius." The experimental results, presented in Table VIII, clearly demonstrate that our proposed methods, with nonzero values for both λ_1 and λ_2, can achieve the best performance in all cases. This indicates that the joint manifold and Frobenius regularization can really boost the multiclass recognition results. Consequently, the optimal values of these two regularization parameters in LDMBLS and REMBLS are not so difficult to be searched, and our proposed methods can



(a) (b) (c) (d)

Fig. 11. Recognition rates versus the selection of λ_1 and λ_2 for LDMBLS on (a) AR and (b) Extended Yale, and REMBLS on (c) AR and (d) Extended YaleB.

achieve satisfactory performances within a large reasonable range.

VI. Conclusion

In this article, two flexible label-induced manifold BLS models, namely LDMBLS and REMBLS, were proposed for multiclass recognition. The strict zero-one label matrix utilized in BLSs was, respectively, relaxed by the proposed label dragging and marginalized retargeted strategy, which can provide more freedom for the regression process and significantly enhance the distinguishability between different classes. The local feature structure was captured to build a class compactness graph, and the trained soft targets were enforced to be consistent with the distribution of data. As the proposed models cannot be solved analytically, two alternating optimization algorithms were derived, in which each step was a convex optimization problem. Thus, the whole iteration process can be scalable and tractable. Extensive experiments on various public databases and solid theoretical analysis verified well the superiority of our proposed methods.

While our proposed algorithms have achieved promising results, there are two future endeavors that deserve attention. First, the class compactness graph in (10) assumes the connected nodes should have the same label. However, the graph edges could contain additional information beyond node similarity. Hence, it would be valuable to explore the integration of effective approaches such as graph convolutional networks [42] and graph attention networks [43] into our proposed models to learn graph embeddings. Second, in practical scenarios, the real datasets are commonly incomplete [44]. Hence, extending the scope of our algorithms to efficiently analyze incomplete data [45] can be another interesting future work.

References

[1] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[2] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3200–3225, Mar. 2023.

[3] L. Bai, Q. Liu, C. Li, C. Zhu, Z. Ye, and M. Xi, "A lightweight and multiscale network for remote sensing image scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[4] Y. Zhang et al., "Dual-constrained deep semi-supervised coupled factorization network with enriched prior," *Int. J. Comput. Vis.*, vol. 129, no. 12, pp. 3233–3254, Dec. 2021.

[5] X. Shi, Q. He, X. Luo, Y. Bai, and M. Shang, "Large-scale and scalable latent factor analysis via distributed alternative stochastic gradient descent for recommender systems," *IEEE Trans. Big Data*, vol. 8, no. 2, pp. 420–431, Apr. 2022.

[6] L. Liu, L. Cai, T. Xie, and Y. Wang, "Self-paced broad learning system," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 4029–4042, Jun. 2023.

[7] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[8] J. Jin, Y. Li, T. Yang, L. Zhao, J. Duan, and C. L. P. Chen, "Discriminative group-sparsity constrained broad learning system for visual recognition," *Inf. Sci.*, vol. 576, pp. 800–818, Oct. 2021.

[9] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.

[10] L. Zhang et al., "Analysis and variants of broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 334–344, Jan. 2022.

[11] S. Wu, J. Wang, H. Sun, K. Zhang, and N. R. Pal, "Fractional approximation of broad learning system," *IEEE Trans. Cybern.*, early access, May 27, 2022, doi: 10.1109/TCYB.2021.3127152.

[12] S. Feng, C. L. P. Chen, L. Xu, and Z. Liu, "On the accuracy–complexity tradeoff of fuzzy broad learning system," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 10, pp. 2963–2974, Oct. 2021.

[13] J. Jin, Y. Li, and C. L. P. Chen, "Pattern classification with corrupted labeling via robust broad learning system," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4959–4971, Oct. 2022.

[14] X. Gong, T. Zhang, C. L. P. Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8922–8950, Sep. 2022.

[15] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Nov. 2012.

[16] X. Zhang, L. Wang, S. Xiang, and C. Liu, "Retargeted least squares regression algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2206–2213, Sep. 2015.

[17] L. Wang and C. Pan, "Groupwise retargeted least-squares regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1352–1358, Apr. 2018.

[18] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.

[19] N. Han et al., "Double relaxed regression for image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 307–319, Feb. 2020.

[20] Y. Zhang, Z. Zhang, J. Qin, L. Zhang, B. Li, and F. Li, "Semi-supervised local multi-manifold isomap by linear embedding for feature extraction," *Pattern Recognit.*, vol. 76, pp. 662–678, Apr. 2018.

[21] G. Du et al., "Graph-based class-imbalance learning with label enhancement," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 20, 2021, doi: 10.1109/TNNLS.2021.3133262.

[22] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by nuclear/$L_{2,1}$-norm regularization for image feature extraction," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1607–1622, Apr. 2017.

[23] W. Rudin, *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1976.

[24] 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.

[25] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 471–478.

[26] 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.

[27] J. Jin, Z. Liu, and C. L. P. Chen, "Discriminative graph regularized broad learning system for image recognition," *Sci. China Inf. Sci.*, vol. 61, no. 11, pp. 112209:1–112209:14, 2018.

[28] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu, "Inter-class sparsity based discriminative least square regression," *Neural Netw.*, vol. 102, pp. 36–47, Jun. 2018.

[29] X. Fang, Y. Xu, X. Li, Z. Lai, W. K. Wong, and B. Fang, "Regularized label relaxation linear regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1006–1018, Apr. 2018.

[30] D. Yu, Q. Kang, J. Jin, Z. Wang, and X. Li, "Smoothing group $L_{1/2}$ regularized discriminative broad learning system for classification and regression," *Pattern Recognit.*, vol. 141, Sep. 2023, Art. no. 109656.

[31] J. Wen et al., "Discriminative regression with adaptive graph diffusion," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 29, 2022, doi: 10.1109/TNNLS.2022.3185408.

[32] Z. Chen, X. Wu, and J. Kittler, "Fisher regularized ε-dragging for image classification," *IEEE Trans. Cognit. Develop. Syst.*, vol. 15, no. 2, pp. 639–650, Jun. 2022.

[33] Y. Wang, Y. Tan, Y. Y. Tang, H. Chen, C. Zou, and L. Li, "Generalized and discriminative collaborative representation for multiclass classification," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 2675–2686, May 2022.

[34] J. Jin, Z. Qin, D. Yu, Y. Li, J. Liang, and C. L. P. Chen, "Regularized discriminative broad learning system for image classification," *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109306.

[35] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[36] A. Martinez and R. Benavente, "The AR face database," CVC Tech. Rep. 24, 1998.

[37] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008.

[38] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[39] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy. Berlin, Germany: Springer, Oct. 2012, pp. 347–360.

[40] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[41] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[42] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[43] T. He, Y. S. Ong, and L. Bai, "Learning conjoint attentions for graph neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2641–2653.

[44] X. Luo, H. Wu, Z. Wang, J. Wang, and D. Meng, "A novel approach to large-scale dynamically weighted directed network representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9756–9773, Dec. 2022.

[45] X. Luo, H. Wu, and Z. Li, "Neulft: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6148–6166, Jun. 2023.

Junwei Jin received the B.S. degree from Ningxia University, Yinchuan, China, in 2013, and the M.S. and Ph.D. degrees from the University of Macau, Macau, China, in 2015 and 2018, respectively.

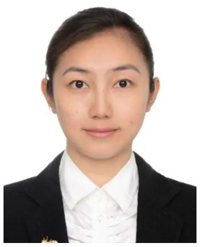
He is currently an Associate Professor with the School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, China. His current research interests include machine learning and neural networks.

Biao Geng is currently pursuing the bachelor's degree with the School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, China.

His current research interests include computer vision, sparse representation, and broad learning.

Yanting Li received the Ph.D. degree from the University of Macau, Macau, China, in 2019.

She is currently an Assistant Professor with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. Her current research interests include function approximation theory and machine learning.



Jing Liang (Senior Member, IEEE) received the B.E. degree from the Harbin Institute of Technology, Harbin, China, in 2003, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2009.

She is currently a Professor with Zhengzhou University, Zhengzhou, China. Her main research interests include evolutionary computation and multiobjective optimization.

Dr. Liang has/had been an Associate Editor of *IEEE Computational Intelligence Magazine*, *Swarm and Evolutionary Computation*, and IEEE Transactions on Evolutionary Computation.

Yang Xiao (Fellow, IEEE) received the B.S. and M.S. degrees in computational mathematics from Jilin University, Changchun, China, in 1989 and 1991, respectively, and the M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH, USA, in 2000 and 2001, respectively.

He is currently a Full Professor with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. His current research interests include cyber-physical systems, wireless networks, and smart grid.

Dr. Xiao currently serves as the Editor-in-Chief for *Cyber-Physical Systems*. He has/had been an Associate Editor for 20 international journals, such as IEEE Transactions on Cybernetics and IEEE Transactions on Systems, Man, and Cybernetics: Systems.

C. L. Philip Chen (Fellow, IEEE) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include cybernetics, systems, and computational intelligence.

Dr. Chen is a fellow of AAAS, IAPR, CAA, and HKIE, and a member of the Academia Europaea (AE), European Academy of Sciences and Arts (EASA), and International Academy of Systems and Cybernetics Science (IASCYS). He is also a Highly Cited Researcher by Clarivate Analytics in 2018 and 2019.