Citation information: DOI: 10.1109/tpami.2026.3655456, IEEE Transactions on Pattern Analysis and Machine Intelligence

Scalable Semi-supervised Learning with Discriminative Label Propagation and Correction

Bingbing Jiang *Member, IEEE*, Jie Wen, *Senior Member, IEEE*, Zidong Wang, *Fellow, IEEE*, Weiguo Sheng, Zhiwen Yu, *Senior Member, IEEE*, Huanhuan Chen, *Fellow, IEEE*, Weiping Ding, *Senior Member, IEEE*

Abstract—**Semi-supervised learning can leverage both labeled and unlabeled samples simultaneously to improve performance. However, existing methods often present the following issues: (1) The emphasis of learning is put on either the similarity structures or the regression losses of data, neglecting the interaction between them. (2) The similarity structures among boundary samples might be unreliable, which misleads label propagation and impairs the performance of models on out-of-sample data. (3) They often involve the inverses of high-order matrices, making them inefficient in computation. To overcome these issues, we propose a scalable semi-supervised learning framework with Discriminative Label Propagation and Correction (DLPC), which collaboratively exploits the regression losses and similarity structures of data. Particularly, each sample is projected onto the independent class labels associated with nonnegative adjustment vectors rather than the propagated labels, such that the distances between samples from different classes are naturally enlarged, making regression losses more effective for boundary samples. Benefiting from this, the regression losses can guide the propagation of labels in boundary areas. Thus, the label information is first propagated through dynamically optimized graph structures and then corrected by the regression losses, effectively improving the quality of labels and facilitating feature projection learning. Furthermore, an accelerated solution has been developed to reduce the computational costs of DLPC on sample scales, thereby making it scalable to relatively large-scale problems. Moreover, the proposed DLPC can not only be applied to single-view scenarios but also extended to multi-view tasks. Additionally, an optimization strategy with fast convergence has been presented for DLPC, and extensive experiments demonstrate the effectiveness and superiority of DLPC over state-of-the-art competitors.**

Index Terms—**Semi-supervised classification, multi-view learning, discriminative label propagation, similarity graph learning.**

I. Introduction

In this information age, there are scarce labeled samples due to the time-consuming process of labeling data, while large amounts of unlabeled samples are easily available in practical applications. As an important learning paradigm, semi-supervised learning (SSL) can exploit both labeled and unlabeled samples simultaneously, obtaining a remarkable advancement [1]–[4]. In the past decades, considerable efforts have been made to develop various SSL methods, such as regression-based methods [5]–[7], graph-based methods [8]–[10], and deep network-based methods [11]–[14]. Among them, graph-based methods construct graphs to capture the similarity relationships among samples, receiving significant attention. Typical graph-based methods explicitly propagate label information from labeled samples to unlabeled samples according to their similarity relationships, enabling neighborhood samples to share similar labels.

Unfortunately, the graph-based label propagation cannot directly handle the samples not used in the training process (i.e., out-of-sample data) [15]. To address this limitation, Nie *et al.* [16] proposed a Flexible Manifold Embedding framework (FME), which introduces the linear regression model to label propagation. Thereafter, many variants of FME have been developed to enhance performance and broaden application domains [10]. However, these methods directly propagate label information through fixed graphs, which completely neglects the interaction between graph structures and label propagation, adversely affecting the quality of propagated labels and learned classifiers. Accordingly, Nie *et al.* [17] proposed a Semi-supervised Flexible Adaptive Graph Embedding (SFAG) framework to dynamically update graph structures during label propagation. Bao *et al.* [18] learned an adaptive graph from the low-rank representation of the original data. To predict training samples, these methods have to calculate the inverses of n-order dense matrices (n is the number of training samples), which requires the computational complexity of $\mathcal{O}(n^3)$ and makes them intractable for large-scale problems. Although some variants attempted to reduce computational costs via constructing bipartite graphs for label propagation, existing graph-based methods still encounter the following limitations. First, the learned graph structures might be unreliable in boundary areas since boundary samples from different classes can be very close to each other, thereby leading to the incorrect propagation of labels through graph structures. Furthermore, they directly utilize the propagated labels as regression targets to train classifiers, such that the low-quality or even incorrect labels mislead the training process, ultimately weakening the effectiveness of classifiers on out-of-sample data.

Bingbing Jiang and Weiguo Sheng are with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China. E-mail: jiangbb@hznu.edu.cn, w.sheng@ieee.org.

Jie Wen is with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Harbin Institute of Technology, Shenzhen 518055, China. E-mail: jiewen_pr@126.com.

Zidong Wang is with the Department of Computer Science, Brunel University London, London, UK. E-mail: zidong.wang@brunel.ac.uk.

Huanhuan Chen is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China. E-mail: hchen@ustc.edu.cn.

Zhiwen Yu is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510650, China. E-mail: zhwyu@scut.edu.cn.

Weiping Ding is with the School of Artificial Intelligence and Computer Science, Nantong University, Nantong, 226019, China, and also the Faculty of Data Science, City University of Macau, Macau 999078, China. E-mail: dwp9988@163.com.

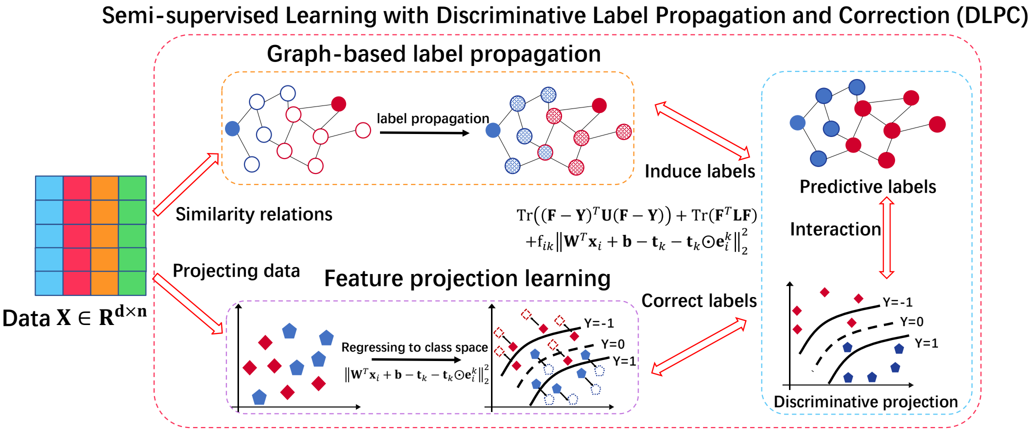

Semi-supervised Learning with Discriminative Label Propagation and Correction (DLPC)
Graph-based label propagation
label propagation
Induce labels
Predictive labels
Similarity relations
$\mathrm{Tr}\big((\mathbf{F}-\mathbf{Y})^T\mathbf{U}(\mathbf{F}-\mathbf{Y})\big)+\mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F})$
$+\mathrm{f}_{ik}\|\mathbf{W}^T\mathbf{x}_i+\mathbf{b}-\mathbf{t}_k-\mathbf{t}_k\odot\mathbf{e}_i^k\|_2^2$
Interaction
Projecting data
Feature projection learning
Data $\mathbf{X}\in\mathbf{R}^{\mathbf{d}\times\mathbf{n}}$
Correct labels
Regressing to class space
$\|\mathbf{W}^T\mathbf{x}_i+\mathbf{b}-\mathbf{t}_k-\mathbf{t}_k\odot\mathbf{e}_i^k\|_2^2$
Y=-1
Y=0
Y=1
Discriminative projection

Fig. 1. The schematic illustration of the proposed DLPC framework, which collaboratively leverages the similarity structures and regression losses of data to propagate and correct the label, so as to reduce the incorrect propagation of labels caused by the unreliable similarity structures on boundary areas and thus enhance the discriminative capability of the feature projection $\mathbf{W}$ for out-of-sample data.

On the other hand, the regression-based methods attempted to evaluate the uncertainties of unlabeled samples via their regression losses on different classes. For example, Wang *et al.* [5] proposed to use the regression losses to evaluate the virtual labels of unlabeled samples and learn a classifier (i.e., feature projection). To enhance the adaptability of regression losses, Luo *et al.* [6] designed a discriminative regression loss, and Qi *et al.* [19] defined a sparse regression loss for semi-supervised learning. In regression-based methods, the regression losses of unlabeled samples are determined by the learned classifier, while the classifier is entirely dependent on labeled samples. Generally, they commonly follow two steps: training a classifier using labeled samples first, and then employing this classifier to evaluate the uncertainties and virtual labels of unlabeled samples, which means that their learning process, in essence, is also to propagate the label information of labeled samples to unlabeled samples. Therefore, the key distinction of regression-based methods to graph-based methods lies in that their medium for propagating label information is the classifier trained on labeled samples, rather than the similarity relationship among samples. Although regression-based methods can suppress the samples with larger uncertainties, their capacity completely depends on labeled samples without actively leveraging the data similarity structures to obtain more label information in the training process, significantly impairing the effectiveness of models.

In recent years, data collected from practical applications are often characterized by multiple feature representations (i.e., views) [20], [21]. Considering that different views are mutually complementary or partly consistent, semi-supervised multi-view learning has attracted growing attention. To explore the similarity structures in different views, graph-based multi-view methods commonly construct graphs on each view and introduce view weights to integrate similarity graphs or label propagation processes on different views. For instance, Li *et al.* [22] proposed to learn a unified graph across multiple views for label propagation, Bahrami *et al.* [23] directly merged the results of label propagation on various views, and Liang *et al.* [24] further distinguished the roles of labeled samples during propagating label information on multiple graphs. Meanwhile, regression-based multi-view methods typically construct regression models on each view to learn view-specific classifiers and combine these classifiers using view weights [25]. For example, Tao *et al.* [26] employed view weights to combine the regression losses of multiple views, and Huang *et al.* [27] derived a shared embedding regularizer for multiple views to learn prediction labels instead of classifiers. Despite achieving progress, graph-based multi-view methods likewise suffer from expensive computation as well as unreliable similarity structures on boundary areas, while regression-based multi-view methods rely entirely on labeled samples and fail to actively enrich label information in training processes, severely limiting the capability of their learned classifiers.

Furthermore, to explore deep representations or graph structures of data, Wang *et al.* [28] learned a deep sparse regularizer for semi-supervised classification, Wu *et al.* [29] established a potential connection between the graph convolutional networks (GCN) and multi-view learning, and Bi *et al.* [30] constructed an additional graph using the node representations of GCN for graph fusion and label propagation. However, these deep network-based methods are designed in a transductive manner [31], focusing on making predictions for the samples used in the training process. Consequently, their trained models or network architectures cannot deal with out-of-sample data, significantly restricting their applicability in practical tasks.

Motivated by the limitations of previous methods, this paper designs a scalable semi-supervised learning framework, i.e., Discriminative Label Propagation and Correction (DLPC), as illustrated in Fig. 1. The main contributions of this paper are summarized as follows:

- We propose a novel semi-supervised learning framework, which effectively bridges the deficiencies of graph-based label propagation and regression models in the process of projection learning, i.e., unreliable label propagation on boundary areas and over-reliance on labeled samples.
- Different from existing methods, the proposed DLPC collaboratively leverages both the regression losses ob-

tained by projecting samples to different classes and the prediction labels propagated through dynamic graph structures to drive semi-supervised learning, not only enhancing the label prediction but also learning a more discriminative classifier for out-of-sample data.

- We devise a multi-view extension for DLPC to adaptively distinguish various views from the aspects of graphs and losses, which can learn a unified graph and the weighted losses compatible across all views, fully preserving the complementarity and correlation among views.
- We develop an optimization strategy to alternatively solve DLPC, and an acceleration solution is further made to significantly reduce the computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2+m^3)$ when tackling large-scale data (n and m denote the numbers of samples and anchors, respectively). Extensive experiments on various datasets demonstrate the outstanding performance of DLPC.

The rest of this paper is organized as follows. Section II reviews related methods. Section III introduces the formulation and optimization procedures of DLPC. Section IV provides the extension and analyses for proposed DLPC. Section V presents extensive experiments to validate DLPC from various perspectives, and Section VI summarizes the conclusion and discusses the limitations of the proposed DLPC as well as potential solutions for future research.

II. Related work

A. Notations

In this paper, matrices are represented by boldface uppercase letters, while vectors are denoted by boldface lowercase letters. For a matrix $\mathbf{R}$, $\mathrm{Tr}(\mathbf{R})$ and $\|\mathbf{R}\|_F$ refer to the trace and the Frobenius norm of $\mathbf{R}$, respectively. $\mathbf{1}_n$ denotes an n-dimensional vector with all-one elements, $\mathbf{I}_n$ denotes an n-order identify matrix, and $\|\mathbf{v}\|_2$ denotes the l_2-norm of a vector $\mathbf{v}$. The feature matrix $\mathbf{X} = [\mathbf{x}_1; \cdots; \mathbf{x}_l; \mathbf{x}_{l+1}; \cdots; \mathbf{x}_n] \in \mathbb{R}^{n\times d}$, where d is the feature dimension, $\mathbf{x}_i|_{i=1}^{l}$ and $\mathbf{x}_j|_{j=l+1}^{n}$ are labeled samples and unlabeled samples, respectively. Define the label matrix $\mathbf{Y} = [\mathbf{Y}_l, \mathbf{Y}_u]^T \in \mathbb{R}^{n\times c}$, in which c is the number of classes, $\mathbf{Y}_l$ and $\mathbf{Y}_u$ denote the given labels of labeled and unlabeled samples, respectively. In the training process, the labels of unlabeled samples are unknown, thus $\mathbf{Y}_u$ is initialized as $\mathbf{0}$.

B. Graph-based Semi-supervised Learning

The graph-based label propagation is a typical semi-supervised learning paradigm, which can be formulated as:

$$\min_{\mathbf{F}} \sum_{i,j=1}^{n} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 s_{ij} + \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T \mathbf{U} (\mathbf{F}-\mathbf{Y}) \right) \tag{1}$$

where $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_n]^T \in \mathbb{R}^{n\times c}$ denotes the virtual label matrix, s_{ij} measures the similarity relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{U}$ is a diagonal matrix, in which the u_{ii} is a large constant (e.g., 10^6) if $\mathbf{x}_i$ is labeled and 0 otherwise. Eq. (1) aims to propagate the label information from labeled samples to unlabeled samples via their similarity relationships and simultaneously makes the prediction labels of labeled samples consistent with the given labels (i.e., $\mathbf{F}_l \approx \mathbf{Y}_l$). Based on Eq. (1), many variants have been proposed, in which the methods closely related to our research are revisited, including the single-view and multi-view semi-supervised models.

(1) *Flexible Manifold Embedding (FME)* [16] introduces the linear regression to the label propagation, and it takes the virtual label $\mathbf{F}$ as a regression target to learn a feature projection subspace, whose objective function is:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{W},\mathbf{b}} & \mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) + \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T \mathbf{U} (\mathbf{F}-\mathbf{Y}) \right) \\ & + \mu \left(\|\mathbf{W}\|_F^2 + \gamma \left\| \mathbf{X}\mathbf{W} + \mathbf{1}_n \mathbf{b}^T - \mathbf{F} \right\|_F^2 \right) \end{aligned} \tag{2}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the graph Laplacian matrix, and $\mathbf{D}$ is the diagonal degree matrix with its i-th diagonal element being $\sum_{j=1}^{n} s_{ij}$. With the learned projection $\mathbf{W} \in \mathbb{R}^{d\times c}$ and bias $\mathbf{b} \in \mathbb{R}^{c\times 1}$, FME can make predictions for out-of-sample data.

(2) *Flexible Multi-view Semi-Supervised Learning (FMSEL)* [22] is a representative variant of FME in the multi-view scenario, which integrates the similarity structures of different views to learn a unified graph $\mathbf{S}$, formulated as:

$$\begin{aligned} \min_{\mathbf{S},\boldsymbol{\alpha},\mathbf{F},\mathbf{W},\mathbf{b}} & \|\mathbf{S} - \sum_{v=1}^{V} \alpha_v \mathbf{A}_v\|_F^2 + \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T \mathbf{U} (\mathbf{F}-\mathbf{Y}) \right) \\ & + \lambda \mathrm{Tr}\left(\mathbf{F}^T\mathbf{L}\mathbf{F} \right) + \beta \left\| \mathbf{X}\mathbf{W} + \mathbf{1}_n \mathbf{b}^T - \mathbf{F} \right\|_F^2 + \xi \|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1}_n = \mathbf{1}_n, \mathbf{S} \geq 0, \boldsymbol{\alpha}^T \mathbf{1}_n = 1, \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{3}$$

where $\{\mathbf{A}_v\}_{v=1}^{V}$ denote the predefined graphs on each view. To make predictions for new samples, FMSEL directly combines the features from different views to learn a concatenated feature projection, ignoring the contribution differences of various views to the feature projection learning.

C. Regression-based Semi-supervised Methods

The regression-based methods focus on evaluating the uncertainties of unlabeled samples via their regression losses on different classes. The representative methods include:

(1) *Adaptive Semi-Supervised Learning (ASL)* employs the predicted labels to characterize the uncertainties of unlabeled samples, whose optimization objective is:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{F}} & \left\| \mathbf{X}_l\mathbf{W} + \mathbf{1}_l \mathbf{b}^T - \mathbf{Y}_l \right\|_F^2 + \sum_{i=1}^{n}\sum_{k=1}^{c} f_{ik}^r \left\| \mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k \right\|_F^2 \\ \text{s.t.} \quad & f_{ik} \in [0,1], \sum_{k=1}^{c} f_{ik} = 1 \end{aligned} \tag{4}$$

where the virtual label f_{ik} evaluates the probability of $\mathbf{x}_i$ belonging to the k-th class, parameter $r \geq 1$ adjusts the distribution of f_{ik}, and $\mathbf{t}_k$ denotes the k-th class indicator.

(2) *Multi-View Semi-Supervised Classification via Adaptive Regression (MVAR)* learns independent classifiers on each view and uses the view weight α_v to combine them, whose objective function is:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{W}_v,\mathbf{b}_v,\alpha_v} & \sum_{v=1}^{V} \alpha_v^\theta \big(\sum_{i=1}^{n} s_i \left\| \mathbf{W}_v^T\mathbf{x}_i^v + \mathbf{b}_v - \mathbf{f}_i \right\|_2 + \lambda_v \|\mathbf{W}_v\|_F^2 \big) \\ \text{s.t.} & \sum_{v=1}^{V} \alpha_v = 1, \alpha_v \geq 0, \mathbf{F}_l = \mathbf{Y}_l, \end{aligned} \tag{5}$$

where λ^v and $\boldsymbol{b}_v$ are the regularization parameter and bias on the v-th view, respectively. The parameter $\theta > 1$ controls the distribution of view weights. The s_i is predetermined to distinguish labeled samples and unlabeled samples (e.g., $s_i = 10^3$ for labeled samples and $s_i = 1$ for unlabeled samples).

III. The proposed Methodology

A. Problem Formulation

To better illustrate the fundamental process by which label information is propagated to unlabeled samples, we first set the derivative of Eq. (1) with respect to $\mathbf{F}$ to zero and obtain:

$$\mathbf{LF} + \mathbf{U}(\mathbf{F} - \mathbf{Y}) = 0 \tag{6}$$

where Eq. (6) can be further reformulated as follows:

$$\left(\begin{bmatrix}\mathbf{D}_{ll} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_{uu}\end{bmatrix} - \begin{bmatrix}\mathbf{S}_{ll} & \mathbf{S}_{lu}\\ \mathbf{S}_{ul} & \mathbf{S}_{uu}\end{bmatrix}\right)\begin{bmatrix}\mathbf{F}_l\\ \mathbf{F}_u\end{bmatrix} + \begin{bmatrix}\mathbf{U}_{ll} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\left(\begin{bmatrix}\mathbf{F}_l\\ \mathbf{F}_u\end{bmatrix} - \begin{bmatrix}\mathbf{Y}_l\\ \mathbf{0}\end{bmatrix}\right) = 0 \tag{7}$$

where $\mathbf{F}_u \in \mathbb{R}^{u\times c}$ are the prediction labels of unlabeled samples, $\mathbf{U}_{ll} \in \mathbb{R}^{l\times l}$, $\mathbf{D}_{ll} \in \mathbb{R}^{l\times l}$ and $\mathbf{D}_{uu} \in \mathbb{R}^{u\times u}$ are the block diagonal matrices of $\mathbf{U}$ and $\mathbf{D}$, respectively. The $\mathbf{S}_{uu} \in \mathbb{R}^{u\times u}$ records the similarity relationships among u unlabeled samples, $\mathbf{S}_{ul} \in \mathbb{R}^{u\times l}$ records the similarity relationships between unlabeled samples and labeled samples, and the same applies to $\mathbf{S}_{ll}$ and $\mathbf{S}_{lu}$. Based on Eq. (7), we can derive the closed-form solution of $\mathbf{F}_u$ as follows:

$$\mathbf{F}_u = (\mathbf{D}_{uu} - \mathbf{S}_{uu} - \mathbf{S}_{ul}\mathbf{K}^{-1}\mathbf{S}_{lu})^{-1}\mathbf{S}_{ul}\mathbf{K}^{-1}\mathbf{U}_{ll}\mathbf{Y}_l \tag{8}$$

where $\mathbf{K} = \mathbf{U}_{ll} + \mathbf{D}_{ll} - \mathbf{S}_{ll}$. The elements in the diagonal matrix $\mathbf{U}_{ll}$ are much larger than those in $\mathbf{D}_{ll}$ and $\mathbf{S}_{ll}$, making $\mathbf{K}^{-1} \approx \mathbf{U}_{ll}^{-1}$. Therefore, the solution of $\mathbf{F}_u$ can be approximately written as:

$$\mathbf{F}_u = (\mathbf{D}_{uu} - \mathbf{S}_{uu} - \mathbf{S}_{ul}\mathbf{U}_{ll}^{-1}\mathbf{S}_{lu})^{-1}\mathbf{S}_{ul}\mathbf{Y}_l \tag{9}$$

In Eq. (9), the elements of $\mathbf{U}_{ll}$ are set as the large const (e.g., 10^6) in advance, while the elements in $\mathbf{S}_{ul}$ and $\mathbf{S}_{lu}$ are within the range of [0,1], such that the elements in $\mathbf{S}_{ul}\mathbf{U}_{ll}^{-1}\mathbf{S}_{lu}$ are much smaller than those in $\mathbf{S}_{uu}$. Meanwhile, the elements of $\mathbf{D}_{uu}$ completely depend on $\mathbf{S}_{uu}$. Therefore, we can conclude that the terms $\mathbf{S}_{uu}$ and $\mathbf{S}_{ul}\mathbf{Y}_l$ play the dominant roles when propagating label information from labeled samples to unlabeled samples. Specifically, the label information in $\mathbf{Y}_l$ is first propagated to the unlabeled samples that have larger similarity relationships with labeled samples via $\mathbf{S}_{ul}\mathbf{Y}_l$, and then further spread to other unlabeled samples through the similarity relationships among unlabeled samples (i.e., $\mathbf{S}_{uu}$).

However, the similarity structures preserved in $\mathbf{S}_{uu}$ might be unreliable in boundary areas since boundary samples belonging to different classes can be quite close. As depicted in Fig. 2(a), there exist several boundary samples that are close to each other (see the points in the black dashed circles). From Fig. 2(b), we observe that the similarity relationships (i.e., the connection lines) among the boundary samples in graph $\mathbf{S}$ are still intensive. The obtained results using Eq. (9) with $\mathbf{S}$ are shown in Fig. 2(c), from which we can find that the unreliable similarity connections on boundary areas indeed misguide the label propagation process, resulting in misclassifications of boundary samples (see the areas in the black dashed circles). Moreover, existing graph-based methods usually use the propagated labels $\mathbf{F} = [\mathbf{F}_l; \mathbf{F}_u]$ as the regression targets to guide the feature projection learning, further impairing the discriminative capability of $\mathbf{W}$ for subsequent tasks.

Motivated by the above analyses and findings (i.e., the unreliable similarity relationships among boundary samples tend to misguide label propagation), we design a novel correction mechanism by collaboratively leveraging the regression losses and similarity structures of data to guide label propagation and projection learning. Specifically, we introduce the independent class label (i.e., indicator variable) $\mathbf{t}_k$ $(k = 1, \cdots, c)$ as the regression target instead of the propagated label $\mathbf{F}$, in which $\mathbf{t}_k = [-1, \cdots, 1, \cdots, -1]^T \in \mathbb{R}^{c\times 1}$ with only the k-th element being 1. Thus, the projected losses of samples on different class indicators are leveraged to correct the labels predicted by label propagation, further alleviating the inaccurate propagation of label information via similarity graphs. To this end, a discriminative semi-supervised label propagation model is designed as follows:

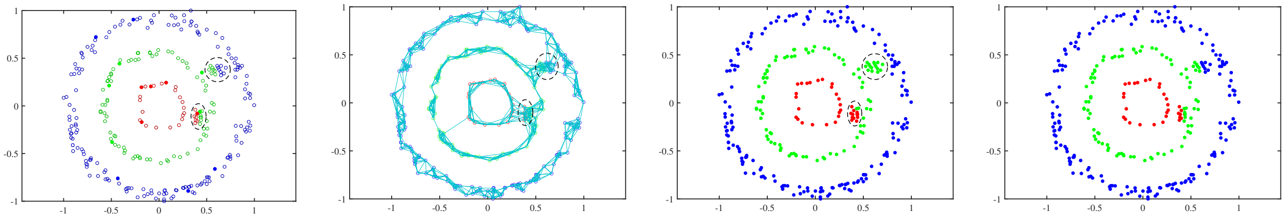
$$\begin{aligned}\min_{\mathbf{W},\mathbf{b},\mathbf{F},\mathbf{e}_i^k}\ & \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T\mathbf{U}(\mathbf{F}-\mathbf{Y})\right) + \mathrm{Tr}(\mathbf{F}^T\mathbf{LF}) + \lambda\|\mathbf{W}\|_F^2\\ & + \varphi\sum_{i=1}^{n}\sum_{k=1}^{c} f_{ik}m_{ik} \quad \text{s.t.} \quad \mathbf{F1}_c = \mathbf{1}_n, \mathbf{F} \geq 0; \mathbf{e}_i^k \geq 0\end{aligned} \tag{10}$$

where $m_{ik} = \left\|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k - \mathbf{t}_k \odot \mathbf{e}_i^k\right\|_2^2$ denotes the regression loss of $\mathbf{x}_i$ on the k-class indicator $\mathbf{t}_k$, and $\odot$ is the Hadamard product operator. In Eq. (10), each sample is projected onto the independent class indicators associated with a nonnegative adjustment vector rather than the possibly incorrect labels f_{ik}, which enlarges the geometrical distances between samples from different classes and makes the obtained regression losses discriminative for the classification boundary. Moreover, the propagated labels and the regression losses can promote each other mutually, not only improving the quality of labels but also learning a more discriminative classifier for out-of-sample data. Specifically, the propagated label f_{ik} acts as the probability of $\mathbf{x}_i$ belonging to the k-th class, such that a high-quality f_{ik} corresponds to a smaller m_{ik} when projecting $\mathbf{x}_i$ to the k-th class, enhancing the effectiveness of regression losses and the feature projection. On the other hand, the regression loss m_{ik} obtained by projecting $\mathbf{x}_i$ to different classes can suppress the samples with larger uncertainties (i.e., boundary samples), such that a larger regression loss m_{ik} corresponds to a smaller probability of $\mathbf{x}_i$ belonging to the k-th class in the projection space, positively guiding the propagation of labels in boundary areas. To further clarify how the regression losses work, we revisit the subproblem of $\mathbf{F}$ in Eq. (10), which is formulated as:

$$\begin{aligned}\min_{\mathbf{F}}\ & \mathrm{Tr}(\mathbf{F}^T\mathbf{LF}) + \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T\mathbf{U}(\mathbf{F}-\mathbf{Y})\right) + \varphi\mathrm{Tr}(\mathbf{MF}^T)\\ \text{s.t.}\ & \mathbf{F1}_c = \mathbf{1}_n, \mathbf{F} \geq 0\end{aligned} \tag{11}$$

where $\mathbf{M} = [\mathbf{M}_l; \mathbf{M}_u] \in \mathbb{R}^{n\times c}$ denotes the regression loss on labeled and unlabeled samples. We first ignore the constraints on $\mathbf{F}$ and set the derivative of Eq. (11) w.r.t. $\mathbf{F}$ to zero, the latent solution of $\mathbf{F}$ can be achieved as follows:

$$\begin{cases}(\mathbf{D}_{ll} - \mathbf{S}_{ll})\mathbf{F}_l - \mathbf{S}_{lu}\mathbf{F}_u + \mathbf{U}_{ll}(\mathbf{F}_l - \mathbf{Y}_l) + \frac{\varphi}{2}\mathbf{M}_l = 0\\ (\mathbf{D}_{uu} - \mathbf{S}_{uu})\mathbf{F}_u - \mathbf{S}_{ul}\mathbf{F}_l + \frac{\varphi}{2}\mathbf{M}_u = 0\end{cases} \tag{12}$$



(a) Original data (b) Constructed graph (c) Results by Eq. (10) (d) Results by Eq. (14)

Fig. 2. (a) shows the original data from three classes, in which filled and hollow points denote the labeled and unlabeled samples, respectively. (b) shows the similarity graph constructed by the k-nearest neighbor ($k = 7$) method, (c) shows the classification result by Eq. (9), and (d) shows the result of Eq. (14).

From the first equation of Eq. (12), we can get $\mathbf{F}_l = (\mathbf{D}_{ll} + \mathbf{U}_{ll} - \mathbf{S}_{ll})^{-1}(\mathbf{S}_{lu}\mathbf{F}_u + \mathbf{U}_{ll}\mathbf{Y}_l - \frac{\varphi}{2}\mathbf{M}_l)$. Then, substituting the solution of $\mathbf{F}_l$ into the second equation of Eq. (12), we obtain the latent solution of $\mathbf{F}_u$ as follows:

$$\begin{aligned}\mathbf{F}_u =&(\mathbf{D}_{uu} - \mathbf{S}_{uu} - \mathbf{S}_{ul}\mathbf{K}^{-1}\mathbf{S}_{lu})^{-1}\\ &\left(\mathbf{S}_{ul}\mathbf{K}^{-1}(\mathbf{U}_{ll}\mathbf{Y}_l - \frac{1}{2}\varphi\mathbf{M}_l) - \frac{1}{2}\varphi\mathbf{M}_u\right)\end{aligned} \tag{13}$$

where $\mathbf{K} = \mathbf{U}_{ll} + \mathbf{D}_{ll} - \mathbf{S}_{ll}$. The elements in the diagonal matrix $\mathbf{U}_{ll}$ are much larger than those in $\mathbf{D}_{ll}$ and $\mathbf{S}_{ll}$, making $\mathbf{K}^{-1}$ ($\mathbf{K} \approx \mathbf{U}_{ll}$) basically close to the zero matrix. Therefore, the solution of $\mathbf{F}_u$ in Eq. (13) can be approximated as:

$$\mathbf{F}_u = (\mathbf{D}_{uu} - \mathbf{S}_{uu} - \mathbf{S}_{ul}\mathbf{U}_{ll}^{-1}\mathbf{S}_{lu})^{-1}(\mathbf{S}_{ul}\mathbf{Y}_l - \frac{1}{2}\varphi\mathbf{M}_u) \tag{14}$$

where the regression loss $\mathbf{M}_u \in \mathbb{R}^{u\times c}$ is incorporated into the label propagation process. As mentioned above, existing graph-based methods often suffer from unreliable $\mathbf{S}_{uu}$ on boundary areas, making propagated labels inaccurate shown in Fig. 2(c). In contrast to the unreliable similarity relationships preserved in $\mathbf{S}_{uu}$, the regression losses obtained by projecting samples to the independent class labels associated with nonnegative adjustment vectors can effectively suppress boundary samples, which are more discriminative for classification boundaries. As a result, the labels are propagated from labeled samples to unlabeled samples first and then be further corrected by regression losses (i.e., the term $\mathbf{S}_{ul}\mathbf{Y}_l - \frac{1}{2}\varphi\mathbf{M}_u$), thereby alleviating the misclassifications on boundary areas in a great extent. Therefore, the proposed correction mechanism in Eq. (11) is essentially a significant improvement and an effective attempt for semi-supervised learning, based on the deep analyses of previous graph-based methods and regression-based methods. The results shown in Fig. 2(d) demonstrate that Eq. (14) correctly predicts almost all boundary samples, greatly improving the quality of labels on boundary areas.

To better facilitate the graph-based label propagation in Eq. (10), graph $\mathbf{S}$ should be timely updated, apart from preserving the similarity relationships among samples. To achieve this, we propose to dynamically learn the graph structure during propagating labels and finally achieve a novel semi-supervised learning framework, formulated as:

$$\begin{aligned}&\min_{\mathbf{W},\mathbf{b},\mathbf{F},\mathbf{S},\mathbf{e}_i^k} \mathrm{Tr}\left((\mathbf{F}-\mathbf{Y})^T\mathbf{U}(\mathbf{F}-\mathbf{Y})\right) + \mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) + \beta\left\|\mathbf{S}-\mathbf{A}\right\|_F^2\\ &+\varphi\sum_{i=1}^{n}\sum_{k=1}^{c} f_{ik}\left\|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k - \mathbf{t}_k\odot\mathbf{e}_i^k\right\|_2^2 + \lambda\left\|\mathbf{W}\right\|_F^2\\ &\text{s.t.}\quad \mathbf{F}\mathbf{1}_c = \mathbf{1}_n, \mathbf{F}\geq 0; \mathbf{e}_i^k \geq 0; \mathbf{S}\mathbf{1}_n = \mathbf{1}_n, \mathbf{S}\geq 0\end{aligned} \tag{15}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a pre-constructed graph using the k-nearest neighbor manner (the details refer to the supplemental material) [32]. Different from graph-based methods and regression models, the proposed DLPC simultaneously leverages the regression losses and similarity structures of data to guide the semi-supervised learning process, which makes up for the deficiencies of previous methods in terms of projection learning and label prediction. Specifically, the regression losses obtained by projecting samples onto the independent class indicators can suppress boundary samples, thereby improving the quality of labels propagated on boundary areas. Moreover, the labels propagated via the dynamically updated graph $\mathbf{S}$ evaluate the probabilities of samples belonging to different classes and will be further optimized with the feature projection, in turn enhancing the effectiveness of regression losses. Benefiting from this, the propagated labels and the regression losses can collaboratively interact with each other, which improves the label prediction and facilitates learning a more discriminative feature projection $\mathbf{W}$ for out-of-sample data.

B. *Optimization Procedures*

Since the objective function of Eq. (15) is not jointly convex w.r.t. all variables, we adopt an alternating optimization strategy to iteratively solve the subproblem on each variable via fixing other variables.

- **Update F:** When other variables are fixed, the subproblem of $\mathbf{F}$ becomes Eq. (11). Thus, the latent solution of $\mathbf{F}_u$ without the constraints can be directly calculated by Eq. (14), denoted as $\widehat{\mathbf{F}}_u$. After that, $\mathbf{F}_u$ can be solved by projecting $\widehat{\mathbf{F}}_u$ to the constrained space:

$$\min_{\mathbf{F}_u\mathbf{1}_c=\mathbf{1}_u,\mathbf{F}_u\geq 0} \left\|\mathbf{F}_u - \widehat{\mathbf{F}}_u\right\|_F^2 \tag{16}$$

Noting that the optimization in Eq. (16) is independent for each row, thus the i-th row of $\mathbf{F}_u$ can be solved by:

$$\min_{\mathbf{f}_{ui}\mathbf{1}_c=1,\mathbf{f}_{ui}\geq 0} \left\|\mathbf{f}_{ui} - \widehat{\mathbf{f}}_{ui}\right\|_2^2 \tag{17}$$

where $\mathbf{f}_{ui}$ and $\widehat{\mathbf{f}}_{ui}$ denote the i-th row of $\mathbf{F}_u$ and $\widehat{\mathbf{F}}_u$, respectively. Eq. (17) is actually the projection problem that projects the row vector $\widehat{\mathbf{f}}_u^i$ onto the probability simplex, whose optimization details can refer to the supplemental material.

- **Update W and b:** With other variables fixed, the optimization problem for $\mathbf{W}$ and $\mathbf{b}$ in Eq. (15) becomes:

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{n}\sum_{k=1}^{c} f_{ik}\left\|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k - \mathbf{t}_k\odot\mathbf{e}_i^k\right\|_2^2 + \lambda\left\|\mathbf{W}\right\|_F^2 \tag{18}$$

Eq. (18) can be rewritten in compact matrix form as:

$$\min_{\mathbf{W},\mathbf{b}} \operatorname{Tr}((\mathbf{X}\mathbf{W}+\mathbf{1}_n\mathbf{b}^T)^T(\mathbf{X}\mathbf{W}+\mathbf{1}_n\mathbf{b}^T)) - 2\operatorname{Tr}(\mathbf{H}(\mathbf{X}\mathbf{W}+\mathbf{1}_n\mathbf{b}^T)^T) + \lambda\operatorname{Tr}(\mathbf{W}^T\mathbf{W}) \tag{19}$$

where $\mathbf{H} \in \mathbb{R}^{n\times c}$ and its i-th row $\mathbf{h}_i$ is calculated as $\sum_{k=1}^{c} f_{ik}\left(\mathbf{t}_k + \mathbf{t}_k \odot \mathbf{e}_i^k\right)$. By setting the derivative of Eq. (19) w.r.t $\mathbf{b}$ to zero, we have:

$$\mathbf{b} = \frac{1}{n}\left(\mathbf{H} - \mathbf{X}\mathbf{W}\right)^T \mathbf{1}_n \tag{20}$$

Then, we set the derivative of Eq. (19) w.r.t $\mathbf{W}$ to zero and substitute $\mathbf{b}$ of Eq. (20) into it, the optimal solution of $\mathbf{W}$ can be achieved as follows:

$$\mathbf{W} = \begin{cases} \left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{P}, & \text{if } d < n \\ \mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T+\lambda\mathbf{I}\right)^{-1}\mathbf{P}, & \text{otherwise} \end{cases}, \tag{21}$$

where $\mathbf{P} = \mathbf{H} - \mathbf{1}_n\mathbf{b}^T \in \mathbb{R}^{n\times c}$. With the optimal solutions of $\mathbf{W}$ and $\mathbf{b}$, we can make predictions for out-of-sample data.

- **Update S:** With other variables fixed except $\mathbf{S}$, the optimization problem for $\mathbf{S}$ can be reformulated as:

$$\min_{\mathbf{S}\mathbf{1}_n=\mathbf{1}_n,\mathbf{S}\geq 0} \frac{1}{2}\sum_{i,j=1}^{n} \|\mathbf{f}_i-\mathbf{f}_j\|_2^2 s_{ij} + \beta\sum_{i=1}^{n}\|\mathbf{s}_i-\boldsymbol{a}_i\|_2^2 \tag{22}$$

where $\mathbf{s}_i$ and $\boldsymbol{a}_i$ denote the i-th rows of $\mathbf{S}$ and $\mathbf{A}$, respectively. Since the problem in Eq. (22) is independent for each row of $\mathbf{S}$, $\mathbf{s}_i$ can be individually solved by:

$$\min_{\mathbf{s}_i\mathbf{1}_n=1,s_{ij}\geq 0} \|\mathbf{s}_i+\mathbf{d}_i\|_2^2 \tag{23}$$

where $\mathbf{d}_i \in \mathbb{R}^{1\times n}$ with $d_{ij} = \frac{1}{4\beta}\|\mathbf{f}_i-\mathbf{f}_j\|_2^2 - a_{ij}$. Eq. (23) can be solved just as Eq. (17).

- **Update $\mathbf{e}_i^k$:** By fixing other variables, we find that $\mathbf{e}_i^k$ $(i=1,\cdots,n; k=1,\cdots,c)$ are independent from each other. Denoting $\mathbf{l}_i^k = \mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k$, Eq. (15) can be disassembled into $n\times c$ subproblems as follows:

$$\min_{\mathbf{e}_i^k\geq 0} \left\|\mathbf{l}_i^k - \mathbf{t}_k \odot \mathbf{e}_i^k\right\|_2^2 \tag{24}$$

Based on the property of squared l_2-norm, Eq. (24) can be decoupled into the following c subproblems:

$$\min_{e_{ij}^k\geq 0} \left(l_{ij}^k - t_{kj}e_{ij}^k\right)^2, \ \ j=1,\cdots,c \tag{25}$$

where e_{ij}^k, l_{ij}^k and t_{kj} are the j-th elements of the $\mathbf{e}_i^k$, $\mathbf{l}_i^k$ and $\mathbf{t}_k$, respectively. Due to $(t_{kj})^2=1$, it is easy to derive that $(l_{ij}^k - t_{kj}e_{ij}^k)^2 = (e_{ij}^k - l_{ij}^k t_{kj})^2$. Considering that e_{ij}^k is nonnegative, we can get optimal solution of e_{ij}^k:

$$e_{ij}^k = \max\left(l_{ij}^k t_{kj}, 0\right) \tag{26}$$

Hence, each $\mathbf{e}_i^k$ can be finally updated by:

$$\mathbf{e}_i^k = \max\left(\mathbf{l}_i^k \odot \mathbf{t}_k, 0\right) \tag{27}$$

The above optimization procedures are iteratively repeated until convergence. Algorithm 1 further summarizes the procedures of solving the objective function in Eq. (15). It should be pointed out that the proposed DLPC follows the inductive learning paradigm [31], thus it can directly use the learned $\mathbf{F}$ and $\mathbf{W}$ to make predictions for unlabeled samples (for training) and testing samples (i.e., out-of-sample data) without additional steps. Specifically, for unlabeled sample $\mathbf{x}_i$ $(l+1\leq i\leq n)$, its class label $y_{\mathbf{x}_i}$ can be predicted by the label vector $\mathbf{f}_i = [f_{i1},\cdots,f_{ic}]^T$, i.e., $y_{\mathbf{x}_i} = \arg\max_{j\leq c} f_{ij}$; for a new sample $\mathbf{x}_{\text{new}}$, we first compute its label vector $\mathbf{o} = \mathbf{W}^T\mathbf{x}_{\text{new}} + \mathbf{b} = [o_1,\cdots,o_c]^T$ using the learned feature projection $\mathbf{W}$ and bias $\mathbf{b}$, and them determine its class label by $y_{\mathbf{x}_{\text{new}}} = \arg\max_{j\leq c} o_j$.

Algorithm 1 : The optimization procedures for Eq. (15)

```
Input: Data $\mathbf{X} \in \mathbb{R}^{n\times d}$ consisting of $l$ labeled and $u$ unlabeled samples, ground truth labels $\mathbf{Y}_l$ of labeled data $\mathbf{X}_l$, initial graph $\mathbf{A}$, parameters $\varphi$, $\lambda$ and $\beta$
Initialize $\mathbf{W}$ and $\mathbf{b}$ by least squares regression on $\{\mathbf{X}_l, \mathbf{Y}_l\}$
repeat
    Update the latent solution of $\mathbf{F}_u$ by Eq. (14);
    Update each row of $\mathbf{F}_u$ by solving Eq. (17);
    Update $\mathbf{b}$ and $\mathbf{W}$ by Eq. (20) and Eq. (21);
    Update each row of $\mathbf{S}$ by solving Eq. (23);
    Update $\mathbf{e}_i^k$ by Eq. (27);
until the objective function of Eq. (15) converges;
Output: The prediction label $\mathbf{F}$, the feature projection $\mathbf{W}$ and the bias term $\mathbf{b}$.
```

IV. The Extension and Analyses of DLPC

A. Multi-View Extension for DLPC

Recently, data generated from real-world applications are often characterized by multiple views, in which each view denotes a distinct feature representation [33]. In this section, we extend the proposed semi-supervised learning framework (i.e., DLPC) to multi-view tasks and develop a multi-view semi-supervised method. In multi-view scenarios, $\mathbf{X} = [\mathbf{X}^1, \mathbf{X}^2\cdots,\mathbf{X}^V]$, and $\mathbf{X}^v \in \mathbb{R}^{n\times d_v}$ represents the v-th view with d_v features, which often contains complementary or partly independent information to other views. To effectively leverage different views, we assign the adaptive weights α_v $(v=1,\cdots,V)$ for each view to preserve both the complementarity and consensus across multiple views in terms of both the regression losses and similarity structures. The objective function of the multi-view extension DLPC is formulated as:

$$\begin{aligned} &\min_{\mathbf{W}^v,\mathbf{b}^v,\mathbf{F},\mathbf{S},\mathbf{e}_i^{v,k}\geq 0,\alpha_v\geq 0} \operatorname{Tr}\left((\mathbf{F}-\mathbf{Y})^T\mathbf{U}(\mathbf{F}-\mathbf{Y})\right) + \operatorname{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) \\ &+ \varphi\sum_{v=1}^{V}\alpha_v\sum_{i=1}^{n}\sum_{k=1}^{c} f_{ik}m_{ik}^v + \lambda\|\mathbf{W}_v\|_F^2 + \beta\Big\|\mathbf{S} - \sum_{v=1}^{V}\alpha_v\mathbf{A}^v\Big\|_F^2 \\ &\text{s.t. } \mathbf{F}\mathbf{1}_c = \mathbf{1}_n, \mathbf{F}\geq 0; \mathbf{S}\mathbf{1}_n = \mathbf{1}_n, \mathbf{S}\geq 0; \sum_{v=1}^{V}\alpha_v = 1, \end{aligned} \tag{28}$$

where $m_{ik}^v = \left\|\mathbf{W}_v^T\mathbf{x}_i^v + \mathbf{b}^v - \mathbf{t}_k - \mathbf{t}_k \odot \mathbf{e}_i^{v,k}\right\|_2^2$ and $\mathbf{A}^v$ denote the regression loss and predefined graph on the v-view, respectively, and $\mathbf{e}_i^{v,k}$ is the nonnegative adjustment vector for $\mathbf{x}_i^v$. Different from previous methods, the multi-view extension of DLPC (i.e., mDLPC) discriminates different views from both the aspects of regression losses and similarity structures, and fuses them with the adaptive view weights $\{\alpha_v\}_{v=1}^V$, fully considering the complementarity of multi-view data in diverse aspects. To ensure the consistency among views, the prediction label f_{ik} is taken as the common target across views, which is jointly determined by the weighted regression losses of sample $\mathbf{x}_i$ on all V views. Moreover, mDLPC explores the compatible similarity structure across views through optimizing the discrepancy between $\mathbf{S}$ and $\{\mathbf{A}^v\}_{v=1}^V$, facilitating the consensus among similarity structures. Benefitting from this way, mDLPC can learn a unified graph and the joint regression losses compatible across multiple views, such that the complementarity and consistency among views are fully leveraged and balanced, positively facilitating label propagation and view-specific projection learning.

Similar to the single-view model in Eq. (15), the variables in Eq. (28) can be solved by Algorithm 1 except for α_v. Specifically, the subproblems of $\mathbf{F}$ and $\mathbf{S}$ are equivalent to those of Eq. (15) when other variables are fixed. Thus, $\mathbf{F}$ and $\mathbf{S}$ can be solved by Eq. (17) and Eq. (23). Since the correlations among views are decoupled, the bias $\mathbf{b}^v$ and the projection $\mathbf{W}^v$ on each view can be separately solved by Eq. (20) and Eq. (21). Besides, the $\mathbf{e}_i^{v,k}$ on each view is likewise independent with each other, thus we can use Eq. (27) to update $\mathbf{e}_i^{v,k} = \max\left(\mathbf{l}_i^{v,k} \odot \mathbf{t}_k, 0\right)$, where $\mathbf{l}_i^{v,k} = \mathbf{W}_v^T \mathbf{x}_i^v + \mathbf{b}^v - \mathbf{t}_k$. Fixing other variables, the optimization problem for $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_V]^T \in \mathbb{R}^{V\times 1}$ is formulated as:

$$\min_{\boldsymbol{\alpha}} \ \|\mathrm{vec}\,(\mathbf{S}) - \boldsymbol{\Gamma}\boldsymbol{\alpha}\|_2^2 + \frac{\boldsymbol{\alpha}^T\boldsymbol{\eta}}{\beta} \ \text{ s.t. } \boldsymbol{\alpha}^T\mathbf{1}_V = 1, \boldsymbol{\alpha} \geq 0 \quad (29)$$

where the $\mathrm{vec}\,(\cdot)$ denotes an operator that converts a matrix into the column vector, $\mathrm{vec}\,(\mathbf{S}) \in \mathbb{R}^{n^2\times 1}$, $\boldsymbol{\Gamma} = \left[\mathrm{vec}\left(\mathbf{A}^1\right), \cdots, \mathrm{vec}\left(\mathbf{A}^V\right)\right] \in \mathbb{R}^{n^2\times V}$, and $\boldsymbol{\eta} \in \mathbb{R}^{V\times 1}$ with its v-th element $\eta_v = \varphi \sum_{i=1}^n \sum_{k=1}^c f_{ik} m_{ik}^v$. Eq. (29) can be further transformed into:

$$\min_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{h} \quad \text{s.t. } \boldsymbol{\alpha}^T\mathbf{1}_V = 1, \boldsymbol{\alpha} \geq 0 \qquad (30)$$

where $\mathbf{h} = \left(2\boldsymbol{\Gamma}^T\mathrm{vec}\,(\mathbf{S}) - \frac{1}{\beta}\boldsymbol{\eta}\right)$. Due to the semi-definite $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$, Eq. (30) is a standard convex quadratic programming (QP) problem, which can be efficiently solved by the Augmented Lagrangian Multiplier (ALM) method (the details can refer to the supplementary material) [34]. In this way, the weight α_v can be adaptively assigned to each view according to the regression losses and similarity structures on different views.

B. Complexity Analysis of DLPC

In Algorithm 1, we optimize Eq. (15) by iteratively solving each variable. Specifically, computing the latent solution $\widehat{\mathbf{F}}u$ using Eq. (14) involves the inverse of an $u \times u$ dense matrix, which requires $\mathcal{O}(u^3)$. Once we have $\widehat{\mathbf{F}}u$, solving $\mathbf{F}_u$ merely takes $\mathcal{O}(uc)$. Updating $\mathbf{W}$ according to Eq. (21) necessitates the inversion of a square matrix, leading to a cost of $\mathcal{O}\,(nd * \min(n, d))$ for each iteration. The optimization of $\mathbf{e}_i^k$ typically requires $\mathcal{O}(nc)$, while updating $\mathbf{b}$ takes a cost of $\mathcal{O}(nc + ndc)$. For $\mathbf{S}$, we need to construct the initial graph $\mathbf{A}$ and update each $\mathbf{s}_i$ by solving Eq. (23), together requiring $\mathcal{O}(ndk + n^2)$ for entire $\mathbf{S}$. Considering that $c \ll n$ and $u \approx n$, the overall computational complexity of our model in each iteration is $\mathcal{O}(n^3 + nd * \min(n, d) + n^2)$.

The optimization of multi-view DLPC in Eq. (28) can be divided into several subproblems, in which $\mathbf{F}_u$, $\mathbf{b}^v$, $\mathbf{W}_v$, $\mathbf{S}$ and $\mathbf{e}_i^{v,k}$ are directly updated by using the corresponding steps in Algorithm 1, approximately requiring $\mathcal{O}\left(n^3 + nd_v c + nd_v * \min(n, d_v) + n^2 + Vnc\right)$. Besides, it usually costs $\mathcal{O}\,(V)$ to update the view weight $\boldsymbol{\alpha}$. Considering $c \ll d_v$ and $c \ll n$, the computational complexity to solve multi-view model is about $\mathcal{O}\left(n^3 + \sum_{v=1}^V (nd_v * \min(n, d_v)) + n^2\right)$.

C. Accelerating Solution for Large-Scale Data

In the optimization procedures, the single-view or multi-view DLPC needs to calculate the inverse of the $u \times u$ dense matrix (i.e., $\mathbf{U}_{uu} + \mathbf{D}_{uu} - \mathbf{S}_{uu} - \mathbf{S}_{ul}\mathbf{K}^{-1}\mathbf{S}_{lu}$), which requires the computation complexity of $\mathcal{O}(u^3)$ ($u \approx n$) and limits its applicability and efficiency for large-scale tasks. To this end, an anchor-based bipartite graph strategy [35] is employed to exploit the neighbor relationships between training samples and anchors, transforming the $n \times n$ full graphs (i.e., $\mathbf{S}$ and $\mathbf{A}$) into $n \times m$ bipartite graph. Concretely, k-means clustering is utilized to generate m ($m \ll n$) cluster centers as anchor points (i.e., $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_m] \in \mathbb{R}^{m\times d}$) from the original data, requiring the computational complexity of $\mathcal{O}\,(nmd)$. The generated anchors can be treated as unlabeled samples, and $\mathbf{G} \in \mathbb{R}^{m\times c}$ denotes their prediction label matrix. Based on the bipartite graph $\mathbf{S}$, an augmented graph $\widehat{\mathbf{S}}$ can be defined as:

$$\widehat{\mathbf{S}} = \begin{bmatrix} \mathbf{0} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)} \qquad (31)$$

Therefore, the single-view DLPC can be reformulated as:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{Q},\mathbf{S},\mathbf{e}_i^k} \ & \mathrm{Tr}\big((\mathbf{Q} - \widehat{\mathbf{Y}})^T\widehat{\mathbf{U}}(\mathbf{Q} - \widehat{\mathbf{Y}})\big) + \mathrm{Tr}(\mathbf{Q}^T\mathbf{L}_{\widehat{\mathbf{S}}}\mathbf{Q}) + \beta\,\|\mathbf{S} - \mathbf{A}\|_F^2 \\ & + \varphi \sum_{i=1}^n \sum_{k=1}^c f_{ik} \left\|\mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{t}_k - \mathbf{t}_k \odot \mathbf{e}_i^k\right\|_2^2 + \lambda\,\|\mathbf{W}\|_F^2 \qquad (32) \\ \text{s.t.} \quad & \mathbf{F}\mathbf{1}_c = \mathbf{1}_n, \mathbf{F} \geq 0; \mathbf{e}_i^k \geq 0; \mathbf{S}\mathbf{1}_m = \mathbf{1}_n, \mathbf{S} \geq 0; \mathbf{G}\mathbf{1}_c = \mathbf{1}_m, \mathbf{G} \geq 0 \end{aligned}$$

where $\mathbf{Q} = \left[\begin{smallmatrix} \mathbf{F} \\ \mathbf{G} \end{smallmatrix}\right] \in \mathbb{R}^{(n+m)\times c}$, $\widehat{\mathbf{Y}} = \left[\begin{smallmatrix} \mathbf{Y} \\ \mathbf{0} \end{smallmatrix}\right] \in \mathbb{R}^{(n+m)\times c}$, $\widehat{\mathbf{U}} = \left[\begin{smallmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right] \in \mathbb{R}^{(n+m)\times(n+m)}$, and $\mathbf{L}_{\widehat{\mathbf{S}}}$ is the Laplacian matrix of $\widehat{\mathbf{S}}$, which is defined as:

$$\mathbf{L}_{\widehat{\mathbf{S}}} = \begin{bmatrix} \mathbf{D}_{\mathbf{S}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{S} \\ -\mathbf{S}^T & \boldsymbol{\Lambda} \end{bmatrix} \qquad (33)$$

where $\mathbf{D}_{\mathbf{S}} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ are the diagonal matrices, whose i-th diagonal elements are the i-th row element sum of $\mathbf{S}$ and the i-th column element sum of $\mathbf{S}$, respectively. Noting that the optimization problems of $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{e}_i^k$ remain unchanged, they can be directly solved through the corresponding steps in Algorithm 1. For the subproblem w.r.t. $\mathbf{S}$, it transforms into:

$$\min_{\mathbf{S}\mathbf{1}_m = \mathbf{1}_n, \mathbf{S} \geq 0} \ \sum_{i=1}^n \sum_{j=1}^m \|\mathbf{f}_i - \mathbf{g}_j\|_2^2\, s_{ij} + \beta \sum_{i=1}^n \|\mathbf{s}_i - \boldsymbol{a}_i\|_2^2 \qquad (34)$$

where $\mathbf{g}_j$ and $\boldsymbol{a}_j$ are the j-th row of $\mathbf{G}$ and $\mathbf{A}$, respectively. Eq. (34) can be solved by rows, requiring the computational complexity of $\mathcal{O}\,(nm)$. To update $\mathbf{F}$ and $\mathbf{G}$, the optimization problem can be written as follows:

$$\begin{aligned}&\min_{\mathbf{F},\mathbf{G}} \operatorname{Tr}\left(\begin{bmatrix}\mathbf{F}-\mathbf{Y}\\ \mathbf{G}\end{bmatrix}^T \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{F}-\mathbf{Y}\\ \mathbf{G}\end{bmatrix}\right)\\ &\quad+\operatorname{Tr}\left(\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}^T\begin{bmatrix}\mathbf{I} & -\mathbf{S}\\ -\mathbf{S}^T & \mathbf{\Lambda}\end{bmatrix}\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}\right)+\varphi\operatorname{Tr}\left(\mathbf{M}\mathbf{F}^T\right)\\ &\text{s.t.}\quad \mathbf{F}\mathbf{1}_c=\mathbf{1}_n,\mathbf{F}\geq 0;\mathbf{G}\mathbf{1}_c=\mathbf{1}_m,\mathbf{G}\geq 0\end{aligned}\tag{35}$$

Taking the derivative of Eq. (35) w.r.t. $\mathbf{G}$ to zero, we have:

$$\mathbf{\Lambda}\mathbf{G}-\mathbf{S}^T\mathbf{F}=0\Longrightarrow\mathbf{G}=\mathbf{\Lambda}^{-1}\mathbf{S}^T\mathbf{F}\tag{36}$$

where the solution of $\mathbf{G}$ also satisfies the constraints of $\mathbf{G}\mathbf{1}_c=\mathbf{1}_m,\mathbf{G}\geq 0$. Substituting Eq. (36) into Eq. (35) and setting the derivative of Eq. (35) w.r.t. $\mathbf{F}$ to zero, we can obtain the latent solution of $\mathbf{F}$ without constraints as follows:

$$\widehat{\mathbf{F}}=\left(\mathbf{U}+\mathbf{I}-\mathbf{S}\mathbf{\Lambda}^{-1}\mathbf{S}^T\right)^{-1}\left(\mathbf{U}\mathbf{Y}-\frac{1}{2}\varphi\mathbf{M}\right)\tag{37}$$

Denoting $\mathbf{O}=\mathbf{U}+\mathbf{I}$ and $\mathbf{V}=\mathbf{U}\mathbf{Y}-\frac{1}{2}\varphi\mathbf{M}$, Eq.(37) can be reformulated according to the matrix identity[1]:

$$\widehat{\mathbf{F}}=\mathbf{O}^{-1}\mathbf{V}+\mathbf{O}^{-1}\mathbf{S}\left(\mathbf{\Lambda}-\mathbf{S}^T\mathbf{O}^{-1}\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{O}^{-1}\mathbf{V}\tag{38}$$

By calculating Eq. (38) from right to left, it avoids the inversion of a high-order matrix and merely needs $\mathcal{O}\left(nm^2+m^3\right)$. Subsequently, it requires $\mathcal{O}\left(nc\right)$ to project $\widehat{\mathbf{F}}$ into the constraint space of $\mathbf{F}\mathbf{1}_c=\mathbf{1}_n,\mathbf{F}\geq 0$. Due to $m\ll n$ and $c\ll n$ for large-scale datasets, the accelerated solution effectively reduces the computational complexity from $\mathcal{O}\big(n^3+nd{*}\min\left(n,d\right)+n^2\big)$ to $\mathcal{O}\big(nm^2+nd{*}\min\left(n,d\right)+m^3\big)$ in each iteration, making DLPC scalable to large-scale tasks.

Meanwhile, the accelerated strategy is likewise applicable to the multi-view DLPC when tackling large-scale multi-view tasks. Considering that anchors generated from different views might be unaligned [36], we first perform k-means on the concatenated views to generate anchors, and split these anchors by views. Noting that the optimization for $\mathbf{W}_v$, $\mathbf{b}^v$, $\mathbf{e}_i^{v,k}$, and $\boldsymbol{\alpha}$ remain unchanged, we can update these variables by the same steps described in Section IV-A. With the predefined bipartite graphs $\{\mathbf{A}^v\}_{v=1}^V$, $\mathbf{S}$ and $\mathbf{F}$ can be updated just as in Eq. (34) and Eq. (38). In this way, the main computational complexity of the multi-view DLPC in Eq. (28) can be reduced from $\mathcal{O}\big(n^3+\sum_{v=1}^V\left(nd_v*\min\left(n,d_v\right)\right)+n^2\big)$ to $\mathcal{O}\big(nm^2+\sum_{v=1}^V\left(nd_v*\min\left(n,d_v\right)\right)+m^3\big)$.

D. Convergence Analysis of DLPC

The variables in the proposed DLPC are alternately optimized, since the objective function of DLPC is not jointly convex with respect to all variables. Therefore, it is necessary to prove that the iterative optimization procedures described in Algorithm 1 can monotonically decrease objective function values in each iteration until convergence. The convergence of the proposed DLPC is proved as follows:

Theorem 1. *The iterative optimization steps described in Algorithm 1 monotonically decrease the objective function value of the proposed DLPC in each iteration until convergence.*

Proof: For convenience, supposing that we have $\mathbf{F}_t$, $\mathbf{W}_t$, $\mathbf{b}_t$, $\mathbf{S}_t$, $\mathbf{e}_{i(t)}^k$, and the objective function value $\ell\big(\mathbf{F}_t,\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)$ at the t-th iteration. When other variables are fixed, in the $t+1$ iteration, $\mathbf{F}$ is updated according to $\mathbf{F}_{t+1}=arg\min_{\mathbf{F}}\ell\big(\mathbf{F}_t,\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)$. Considering that this minimization problem of $\mathbf{F}$ in Eq. (11) or Eq. (35) is convex, we naturally have:

$$\ell\big(\mathbf{F}_{t+1},\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)\leq\ell\big(\mathbf{F}_t,\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big).\tag{39}$$

Fixing other variables, the variables $\mathbf{W}$ and $\mathbf{b}$ are alternately updated by solving the convex subproblem in Eq. (19), i.e., $\left(\mathbf{W}_{t+1},\mathbf{b}_{t+1}\right)$ $=arg\min_{\mathbf{W},\mathbf{b}}\ell\big(\mathbf{F}_{t+1},\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)$. Therefore, we directly get:

$$\begin{aligned}&\ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)\\ \leq&\ell\big(\mathbf{F}_{t+1},\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big).\end{aligned}\tag{40}$$

In the same way, the graph $\mathbf{S}$ is updated according to $\mathbf{S}_{t+1}$ $=arg\min_{\mathbf{S}}\ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)$, thus we have:

$$\begin{aligned}&\ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_{t+1},\mathbf{e}_{i(t)}^k\big)\\ \leq&\ell\big(\mathbf{F}_{t+1},\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_{t+1},\mathbf{e}_{i(t)}^k\big).\end{aligned}\tag{41}$$

Finally, the adjustment variable $\mathbf{e}_i^k$ is updated by $\mathbf{e}_{i(t+1)}^k$ $=arg\min_{\mathbf{e}_i^k}\ell\left(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_{t+1},\mathbf{e}_{i(t)}^k\right)$. Thus, we directly infer that:

$$\begin{aligned}&\ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_{t+1},\mathbf{e}_{i(t+1)}^k\big)\\ \leq&\ \ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_{t+1},\mathbf{e}_{i(t)}^k\big)\end{aligned}\tag{42}$$

Following the above inequalities, we finally have:

$$\ell\big(\mathbf{F}_{t+1},\mathbf{W}_{t+1},\mathbf{b}_{t+1},\mathbf{S}_{t+1},\mathbf{e}_{i(t+1)}^k\big)\leq\ell\big(\mathbf{F}_t,\mathbf{W}_t,\mathbf{b}_t,\mathbf{S}_t,\mathbf{e}_{i(t)}^k\big)\tag{43}$$

Considering that $\ell\big(\mathbf{F},\mathbf{W},\mathbf{b},\mathbf{S},\mathbf{e}_i^k\big)$ is bounded below (at least above 0), we conclude that the objective function of DLPC is monotonously decreased in each iteration until convergence. Notably, the objective function of DLPC can rapidly converge to a stable value within just a few iterations (see the experimental results in Section V-F).

In the multi-view extension of DLPC, the variables can be directly solved by the optimization steps described in Algorithm 1 except for the view weight $\boldsymbol{\alpha}$. We note that the optimization of $\boldsymbol{\alpha}$ in Eq. (30) is a standard convex QP problem, which can be solved with convergence by the ALM method [34]. Therefore, the objective function of multi-view DLPC is also monotonically decreasing until convergence.

V. Experiments

To validate the proposed DLPC framework, extensive experiments have been conducted on various datasets. Firstly, we perform experiments on six benchmarks for semi-supervised learning, including the G241c, G241d, Digit1, USPS, COIL$_2$, and BCI datasets [4]. All six benchmarks have 2 classes, 1500 samples, and 241 features; except for BCI, which has 400 samples and 114 features. For more information about these datasets, please refer to [4]. Secondly, eight (single-view) real-world datasets (including five common datasets and three large-scale datasets) are used to further verify the performance and superiority of DLPC over other methods.

[1] $(\mathbf{A}+\mathbf{C}\mathbf{B}\mathbf{C}^T)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{C}(\mathbf{B}^{-1}+\mathbf{C}^T\mathbf{A}^{-1}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{A}^{-1}$

TABLE I
THE DETAILED INFORMATION OF SINGLE-VIEW DATASETS.

Dataset	DNA [37]	AR [38]	Binalpha[2]	Isolet[3]
Samples	1186	1680	1404	1560
Features	180	1024	320	617
Classees	3	120	36	26
Dataset	Coil20[4]	Connect-4[5]	MNIST [3]	SUSY [31]
Samples	1440	67557	70000	5000000
Features	1024	126	784	18
Classees	20	3	10	2

TABLE II
THE DETAILED INFORMATION OF MULTI-VIEW DATASETS.

Dataset	Classes	Samples	Features
MSRC-v1 [39]	7	210	2418(1302/48/512/100/256/200)
ORL [21]	40	400	1689(512/59/864/254)
BDGP [40]	5	2500	1750(1000/500/250)
Scene15 [41]	15	4485	4420(1800/1180/1240)
Mfeat [11]	10	2000	649(216/76/64/6/240/47)
Hdigt [22]	10	10000	1040(784/256)
VoxCeleb[6]	50	18354	1458(192/512/160/80/514)
Youtube[7]	31	95000	3350(2000/1000/512/838)

Table I summarizes these single-view datasets. Furthermore, experiments on eight multi-view datasets with different data sizes are conducted to evaluate the effectiveness and adaptability of the multi-view DLPC in practical tasks. The details of multi-view datasets are provided in Table II.

A. Experimental Setting

To verify the effectiveness of DLPC, we compare it with the state-of-the-art single-view competitors, including the graph-based methods, i.e., Flexible Manifold Embedding (FME) [16], Adaptive Pairwise Graph Embedding (APGE) [9], Robust Embedding Regression (RER) [18], Semi-supervised Flexible Adaptive Graph (SFAG) [17], Fast Flexible Manifold Embedding (FFME) [42] and Reduced Flexible Manifold Embedding (RFME) [43], and the regression-based methods, i.e., Adaptive Semi-supervised Learning (ASL) [5] and Adaptive semi-supervised learning with Discriminative Least Squares Regression(ADLSR) [6]. Among the above methods, ASL and ADLSR can run on small and large-scale datasets, FFME and RFME are designed for large-scale datasets (i.e, Connect-4, MNIST, and SUSY), and other compared methods can merely deal with small-scale datasets. Meanwhile, to validate the performance on multi-view tasks, the multi-view DLPC (i.e., mDLPC) is compared with state-of-the-art multi-view competitors, including three graph-based methods, i.e., Flexible Multi-view Semi-supervised Learning (FMSEL) [22], Auto-weighted Multi-view Semi-supervised Learning (AMSSL) [23], and Label-Weighted Graph-based Learning (LWGL) [24], three regression-based methods, i.e., Multi-view Semi-supervised Classification via Adaptive Regression (MVAR) [26], Joint Consensus and Diversity (JCD) [39], and Embedding Regularizer Learning (ERL) [27], as well as two deep network-based methods, i.e., Interpretable Multi-view Graph Convolutional Network (IMVGCN) [29] and Sample-weighted Fused Graph-based Semi-supervised Classification (WFGSC) [30]. Among these multi-view methods, LWGL, ERL, and two network-based methods are designed in a transductive manner, which cannot predict testing samples.

TABLE III
THE DETAILED DATA PARTITIONS ON LARGE-SCALE DATASETS.

Dataset	Connect-4	MNIST	SUSY
Samples	67557	70000	5000000
Training Ratio	0.5	0.5	0.1
Anchor Points	300	300	1000
Dataset	Hdigt	Voxceleb	Youtube
Samples	10000	18354	95000
Training Ratio	0.8	0.5	0.2
Anchor Points	300	300	500

For the benchmarks, each dataset contains 12 subsets, in which there are 10 and 100 labeled samples, respectively, and the remaining samples are unlabeled. For other real-world datasets with different data sizes, we divide each dataset into labeled, unlabeled, and testing subsets. Specifically, for the datasets containing less than 10000 samples, we randomly choose 80% of samples as the training subset, and the remaining are used for testing. Among the training subsets, $10\%, 20\%$, and 30% of training samples are assigned with the true class labels, and others are unlabeled samples. For the datasets containing over 10,000 samples, the acceleration solution designed in Section IV-C is used to generate anchors and construct bipartite graphs, in which the generated anchors are likewise applied for other methods (i.e., FFME and RFME). The specific data partitions and the number of anchors for large-scale datasets are reported in Table III. On these large-scale datasets, 1%, 2%, and 3% of training samples are randomly labeled, and the others are unlabeled samples. For a fair comparison, the parameters of all compared methods are tuned using their default settings. In the proposed DLPC, we tune λ and β from $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$ and search φ in the range of $\{0.001, 0.01, 0.1, 1\}$. To reduce the statistical variability of results, all methods are independently run 20 times on disparate training and testing subsets except for the benchmarks, and the average results with optimal parameter configurations are used to evaluate their performance.

B. Experimental Results and Analyses

1) Comparisons on Single-view Datasets: Following the experimental setups in [4], the average accuracies of different methods on six benchmarks are reported in Table IV. As can be seen, the proposed DLPC achieves highly competitive results on all the benchmarks. In particular, DLPC achieves obvious improvements than graph-based and regression-based methods on the G241c dataset, in which boundary samples belonging to different classes are quite close and even overlapped. This demonstrates the effectiveness and competitiveness of the proposed semi-supervised learning framework.

Table V and Table VI report the classification results of different methods for the unlabeled samples and testing samples of five small datasets, and three large-scale datasets, in which the ratio of labeled samples varies. From the experimental results, we find that the graph-based methods (e.g., RER, APGE, SFAG) and the regression-based methods (e.g., ADLSR and ASL) are merely effective on parts of the datasets since they

TABLE IV
TRANSDUCTIVE CLASSIFICATION RESULTS (ACC% ± STD×10^2) OF DIFFERENT METHODS ON SIX BENCHMARKS.

Labeled	Datasets	ADLSR	ASL	FME	APGE	RER	SFAG	DLPC
$l = 10$	G241c	52.36±3.26	52.37±2.89	58.30±2.89	53.04±2.29	56.27±1.53	60.91±1.57	**73.65**±1.34
	G241d	54.60±2.27	54.23±1.72	58.93±2.77	54.95±3.41	55.80±3.34	59.53±3.32	**59.75**±1.84
	Digit1	74.31±1.37	74.34±1.12	83.08±2.31	72.83±1.02	**86.97**±1.75	70.64±1.89	86.09±1.13
	USPS	80.65±2.03	80.40±1.98	83.79±1.66	81.33±0.99	83.16±1.71	75.86±0.51	**83.91**±0.62
	COIL_2	56.25±3.36	56.22±2.96	60.98±4.73	56.66±3.77	58.83±1.17	**61.91**±2.41	59.67±2.30
	BCI	53.57±2.34	53.85±1.29	57.84±2.60	53.31±2.13	58.93±1.61	**59.82**±1.33	58.88±1.79
$l = 100$	G241c	81.37±2.55	80.33±2.45	74.57±3.51	71.86±1.79	72.48±1.66	75.34±1.18	**86.05**±3.21
	G241d	73.75±2.70	74.26±1.58	73.35±2.69	71.69±1.82	73.89±2.52	75.79±1.69	**76.14**±2.55
	Digit1	90.91±1.19	91.16±0.78	94.48±1.69	91.70±0.98	**95.53**±1.55	90.19±2.24	94.71±2.53
	USPS	85.98±1.38	86.49±1.62	**93.34**±0.75	87.37±0.02	86.18±0.25	87.17±0.48	92.98±0.34
	COIL_2	80.69±2.94	80.70±1.80	90.89±2.01	83.24±2.26	89.17±2.68	**91.46**±2.29	91.06±2.05
	BCI	73.17±3.71	72.58±1.63	73.14±2.56	71.92±2.47	**73.85**±1.54	72.97±1.59	73.33±2.39

TABLE V
THE CLASSIFICATION RESULTS (ACC% ± STD×10^2) ON SINGLE-VIEW DATASETS WITH DIFFERENT LABELED RATIOS IN TRAINING SAMPLES.

				Performance on Unlabeled Samples				
Datasets	Labeled Ratio	ADLSR	ASL	FME	APGE	RER	SFAG	DLPC
DNA	0.1	84.50±2.76	83.49±2.26	76.56±2.43	80.42±2.42	82.89±2.72	75.04±3.04	**87.52±2.61**
	0.2	86.40±2.41	86.34±2.37	81.82±1.31	84.35±1.80	85.97±2.52	83.90±2.52	**89.23±1.90**
	0.3	89.24±1.31	89.32±1.47	84.25±1.22	87.28±1.54	87.59±1.78	88.15±1.46	**91.37±1.83**
AR	0.1	75.37±3.66	77.94±3.76	75.95±3.52	78.75±3.71	77.35±3.47	77.12±3.55	**80.20±3.64**
	0.2	88.16±3.68	88.35±5.10	88.06±3.28	88.75±3.54	88.19±2.93	89.34±3.28	**90.73±3.72**
	0.3	90.01±4.51	89.85±5.18	90.26±4.10	90.52±4.79	90.15±4.67	90.25±2.72	**92.46±4.17**
Binalpha	0.1	45.47±2.93	45.21±2.35	**52.61±1.84**	45.59±2.02	47.50±1.73	50.62±1.57	52.12±1.51
	0.2	52.48±2.12	51.57±1.27	59.78±1.73	53.45±1.64	57.09±1.88	57.38±1.31	**60.23±1.75**
	0.3	57.08±2.08	55.89±1.36	63.09±1.18	57.44±1.78	62.57±1.54	62.71±1.09	**64.73±1.44**
Isolet	0.1	85.74±1.87	85.30±1.58	84.35±1.27	86.59±2.45	85.24±1.88	84.94±1.65	**88.81±1.85**
	0.2	89.58±1.52	89.37±1.66	88.53±1.52	90.15±1.12	89.66±1.59	88.47±1.48	**91.83±1.54**
	0.3	90.44±1.23	90.89±0.91	90.03±0.88	91.67±1.04	91.18±1.26	90.83±1.23	**93.14±1.01**
Coil20	0.1	86.84±3.20	86.72±1.97	91.06±1.64	91.02±1.12	91.11±2.36	87.96±2.24	**91.50±2.16**
	0.2	91.94±2.43	91.22±1.29	94.77±1.22	94.37±0.95	94.51±1.17	92.28±1.73	**95.18±1.64**
	0.3	93.43±2.60	92.13±1.36	95.37±0.76	95.20±0.66	95.18±1.23	93.61±1.31	**96.18±1.25**
				Performance on Testing Samples				
Datasets	Labeled Ratio	ADLSR	ASL	FME	APGE	RER	SFAG	DLPC
DNA	0.1	83.94±2.58	83.60±3.39	76.26±2.91	80.99±3.18	79.32±2.94	78.17±2.65	**86.76±2.41**
	0.2	85.19±2.25	84.47±3.03	82.02±2.38	83.87±2.06	84.15±2.40	83.03±1.90	**89.82±2.15**
	0.3	89.48±1.59	91.64±1.38	85.34±2.22	88.61±2.35	88.67±1.98	88.36±1.98	**91.79±1.97**
AR	0.1	77.35±3.34	76.82±3.03	73.69±2.67	77.85±2.72	**79.67±2.70**	77.34±3.29	79.48±3.68
	0.2	88.20±5.19	87.44±5.31	88.61±2.54	88.64±3.24	89.05±2.36	88.25±3.51	**89.55±3.94**
	0.3	89.15±5.11	89.08±4.89	89.49±4.25	89.76±4.35	90.23±3.33	89.78±3.62	**91.47±3.27**
Binalpha	0.1	45.80±1.35	45.73±2.07	46.65±2.06	46.75±2.32	45.72±2.36	45.91±2.04	**46.91±2.23**
	0.2	52.99±2.12	52.78±1.78	52.74±1.55	53.47±2.66	53.25±1.97	53.22±1.35	**54.56±1.48**
	0.3	56.05±2.50	55.59±1.95	56.65±2.50	56.88±2.31	57.18±2.43	57.09±1.72	**59.54±1.72**
Isolet	0.1	86.63±1.88	85.96±2.83	84.26±2.25	86.32±2.97	85.63±1.16	85.87±1.58	**87.53±2.09**
	0.2	89.47±0.79	88.59±1.67	88.94±1.59	89.61±1.29	89.42±1.53	88.65±1.26	**91.83±1.15**
	0.3	90.59±0.96	90.45±1.54	90.42±1.49	90.84±1.82	91.67±1.55	89.72±1.03	**93.11±0.92**
Coil20	0.1	85.24±2.26	84.90±2.84	89.10±3.09	89.85±1.84	89.14±3.24	88.24±2.33	**91.88±1.73**
	0.2	88.86±1.40	89.34±2.46	91.39±1.77	91.22±1.66	91.75±1.77	91.36±1.82	**93.46±1.53**
	0.3	90.91±1.36	90.39±1.55	92.57±1.95	92.80±1.13	92.94±1.39	92.34±1.32	**95.24±1.47**

only consider the data similarity structures or the losses of projecting samples to different classes in their training processes. Taking the DNA and Coil20 datasets as examples, the graph-based methods achieve better performance on Coil20 but perform worse on DNA, while the performance of regression-based methods exhibits the opposite situation. In contrast, the proposed DLPC incorporates the projection losses into the label propagation process, such that the prediction labels are first propagated via dynamically updated graph structures and then further corrected by the losses, alleviating the misclassifications for boundary samples to a great extent. As shown in Table V and Table VI, DLPC demonstrates superior or highly comparable performance across all datasets over state-of-the-art competitors. This validates that the collaborative interaction between similarity structures and regression losses indeed facilitates learning more predictive labels for unlabeled samples and effective classifiers for testing samples.

Fig. 3 further shows the runtime of DLPC using the acceleration solution in Section IV-C and other methods on three large-scale datasets, in which FFME and RFME are the scalable variants of FME. From Tables V, VI, and Fig. 3, we observe that DLPC not only demonstrates better classification accuracy but also achieves promising efficiency when handling these large-scale datasets. This demonstrates that the proposed acceleration solution indeed reduces the computational time of DLPC while maintaining comparable performance.

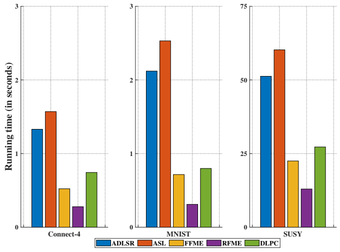


Fig. 3. The runtime of DPLC and other methods on large-scale datasets.

2) Comparisons on Multi-View Datasets: Tables VII and VIII record the classification results of mDLPC and other methods on multi-view datasets, where 'OM' denotes the out-

TABLE VI

THE CLASSIFICATION RESULTS (ACC% $\pm$ STD$\times 10^2$) OF DLPC AND OTHER SCALABLE METHODS ON THREE LARGE-SCALE SINGLE-VIEW DATASETS.

Datasets	Method	1% labeled samples		2% labeled samples		3% labeled samples	
		Unlabeled samples	Testing samples	Unlabeled samples	Testing samples	Unlabeled samples	Testing samples
Connect-4	ADLSR	60.22±3.90	60.25±4.20	64.07±3.42	63.66±3.43	64.42±2.96	63.68±2.78
	ASL	60.32±4.17	60.78±4.48	64.04±3.75	63.63±4.01	64.21±2.85	63.54±2.70
	FFME	56.29±2.15	56.25±2.71	59.26±0.76	59.09±1.41	60.88±1.32	60.58±1.29
	RFME	52.25±1.33	52.05±1.55	57.52±1.84	57.07±1.79	58.28±0.96	58.02±1.01
	DLPC	**62.48±2.08**	**62.28±1.79**	**66.67±1.42**	**65.97±1.51**	**68.86±1.24**	**67.91±0.76**
MNIST	ADLSR	67.54±2.38	66.70±2.50	69.52±0.77	68.59±0.95	70.75±1.01	69.90±0.81
	ASL	65.09±2.50	64.31±2.40	66.69±0.84	65.96±0.75	68.12±0.76	67.36±0.64
	FFME	77.35±1.16	73.48±1.60	80.92±1.18	78.59±1.14	82.02±1.18	80.39±1.21
	RFME	76.31±2.87	72.61±1.25	80.32±2.51	75.36±1.20	82.34±2.49	78.11±1.59
	DLPC	**79.57±1.25**	**78.15±1.20**	**81.99±1.08**	**80.83±1.11**	**84.21±0.89**	**81.66±0.76**
SUSY	ADLSR	54.23±1.19	54.24±1.37	55.17±1.22	55.46±1.06	56.89±1.13	56.19±0.97
	ASL	54.23±1.11	54.23±1.05	54.23±1.08	54.23±0.86	54.23±0.79	54.23±0.62
	FFME	74.31±2.44	74.98±2.65	74.90±2.43	74.68±2.35	75.31±2.29	75.09±2.10
	RFME	73.49±2.59	64.67±2.58	73.97±2.67	72.23±2.46	74.33±2.55	73.69±2.42
	DLPC	**76.81±1.40**	**77.07±1.45**	**77.81±1.28**	**78.07±1.59**	**78.95±1.22**	**78.07±1.35**

TABLE VII

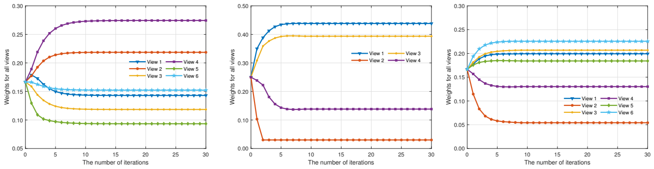
THE CLASSIFICATION RESULTS (ACC% $\pm$ STD$\times 10^2$) ON UNLABELED DATA OF MULTI-VIEW METHODS WITH DIFFERENT LABELED RATIOS.

Datasets	Labeled Ratio	FMSEL	AMSSL	LWGL	ERL	JCD	MVAR	IMVGCN	WFGSC	DLPC
MSRC-v1	0.1	89.67±2.78	89.87±2.69	87.40±2.23	86.43±3.40	88.90±3.07	86.62±3.15	84.53±2.02	86.49±2.75	**91.04±2.49**
	0.2	95.68±2.22	92.03±2.56	88.12±2.06	92.63±2.53	94.66±2.52	93.68±3.02	86.84±1.96	89.16±2.13	**96.69±2.21**
	0.3	97.22±2.53	94.12±2.49	88.57±2.94	93.11±3.10	97.14±2.10	96.64±2.13	88.40±1.18	93.18±2.67	**98.24±2.23**
ORL	0.1	90.07±2.19	89.25±1.89	83.50±2.12	83.79±2.12	83.39±2.72	82.43±3.13	78.21±3.70	82.71±2.26	**90.64±2.17**
	0.2	94.79±0.96	95.17±1.61	93.04±2.06	88.58±2.47	93.38±2.48	92.96±2.32	89.58±3.47	91.02±2.38	**95.54±2.09**
	0.3	97.50±1.03	96.75±1.58	93.58±2.61	94.04±2.24	96.90±2.34	96.75±2.29	93.40±2.82	94.40±1.77	**98.15±1.68**
BDGP	0.1	82.38±0.99	79.93±1.52	70.37±1.67	77.07±1.56	82.47±1.67	**83.57±1.33**	69.46±0.94	75.58±1.60	83.06±1.21
	0.2	85.09±0.90	84.74±1.47	76.41±1.83	78.49±1.09	85.39±1.54	85.37±1.97	75.03±0.67	78.97±1.39	**85.79±1.07**
	0.3	86.59±0.68	86.79±1.33	83.47±1.01	81.43±0.97	86.89±1.18	87.02±1.42	80.15±0.52	81.59±1.20	**87.28±0.98**
Scene15	0.1	64.22±2.92	60.47±2.76	58.09±0.73	59.30±1.37	**64.86±1.97**	57.34±2.34	52.57±1.79	57.56±1.29	64.24±1.20
	0.2	68.96±1.47	64.35±1.03	64.23±0.87	61.50±1.33	69.03±1.22	59.35±1.34	56.65±1.53	61.04±0.67	**69.61±1.05**
	0.3	70.81±1.11	66.65±0.75	67.75±0.96	71.03±1.39	**71.91±0.86**	60.09±1.50	64.87±0.46	67.48±0.58	71.38±0.87
Mfeat	0.1	97.36±2.76	97.19±2.10	96.94±2.08	94.47±1.77	96.69±3.01	96.45±1.66	95.69±0.87	96.22±2.04	**97.90±1.43**
	0.2	98.40±1.11	98.26±1.42	97.06±1.27	95.87±1.28	97.31±1.85	97.19±1.48	96.64±0.33	96.84±1.79	**98.83±1.30**
	0.3	98.66±1.06	98.73±0.77	97.60±1.42	96.28±1.53	97.38±1.29	97.31±1.42	96.84±0.33	97.36±1.18	**98.95±1.21**
Hdigt	0.01	95.73±1.43	95.90±2.03	93.61±1.39	94.55±2.31	96.01±1.28	92.09±0.59	88.73±0.56	90.42±2.13	**96.59±1.35**
	0.02	97.58±1.68	97.45±1.07	96.29±2.15	97.12±1.98	96.13±0.78	93.01±0.46	90.89±0.22	94.56±1.08	**98.49±0.87**
	0.03	98.03±1.06	97.86±1.29	96.85±1.13	97.77±2.16	96.36±0.56	93.73±0.29	93.20±0.37	97.68±0.83	**98.52±0.45**
VoxCeleb	0.01	58.37±0.97	57.19±1.58	56.35±2.58	57.39±1.66	62.48±1.79	61.04±1.93	57.46±1.81	57.93±2.20	**65.40±1.94**
	0.02	73.02±0.68	70.15±1.42	70.77±2.36	73.85±1.78	78.77±1.48	69.97±1.26	71.15±1.19	72.85±0.85	**80.62±1.56**
	0.03	82.03±0.42	81.53±1.23	75.16±2.62	82.11±1.53	85.44±0.68	81.59±0.75	79.38±0.65	81.85±0.94	**87.75±1.12**
Youtube	0.01	OM	OM	OM	OM	56.85±0.19	66.39±0.56	54.07±0.53	58.57±1.12	**70.88±0.79**
	0.02	OM	OM	OM	OM	64.85±0.22	71.97±0.71	60.29±0.46	65.93±0.68	**73.51±0.73**
	0.03	OM	OM	OM	OM	68.09±0.25	74.43±0.64	64.84±0.39	67.96±0.68	**78.05±0.64**

TABLE VIII

THE CLASSIFICATION RESULTS (ACC% $\pm$ STD$\times 10^2$) ON TESTING SAMPLES OF MULTI-VIEW METHODS WITH DIFFERENT LABELED RATIOS.

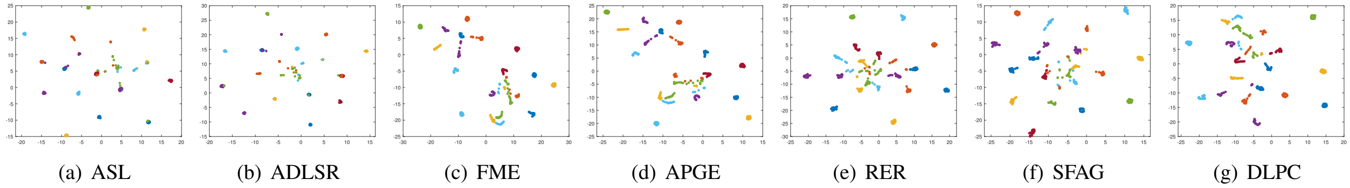
Datasets	Labeled Ratio	FMSEL	AMSSL	JCD	MVAR	mDLPC
MSRC-v1	0.1	87.38±3.83	**92.14±3.76**	90.95±3.55	91.43±4.24	91.52±3.01
	0.2	95.00±3.63	96.19±3.13	97.86±3.37	96.42±3.28	**98.81±2.96**
	0.3	97.63±3.40	96.90±2.57	98.33±3.02	98.10±3.17	**98.81±2.77**
ORL	0.1	90.04±2.79	90.35±2.49	83.38±3.17	85.25±3.49	**90.75±2.89**
	0.2	96.46±1.45	96.08±2.26	95.00±2.89	95.88±3.23	**96.63±2.69**
	0.3	98.54±1.67	**98.75±2.14**	97.88±2.76	98.13±2.43	98.33±2.18
BDGP	0.1	82.70±1.30	80.06±1.68	81.64±1.59	82.38±1.84	**83.24±1.84**
	0.2	84.80±1.73	84.58±1.59	84.72±1.91	84.14±2.11	**85.05±1.59**
	0.3	86.46±1.52	86.56±1.37	86.42±1.80	85.10±2.04	**87.42±1.60**
Scene15	0.1	64.26±2.12	62.12±2.56	**65.81±2.37**	56.12±1.62	64.59±1.12
	0.2	68.23±1.73	65.35±1.48	69.01±1.47	58.22±1.89	**69.30±1.08**
	0.3	70.43±1.42	68.57±1.51	**71.39±1.35**	59.62±1.66	70.94±0.94
Mfeat	0.1	97.30±2.55	97.00±1.29	96.00±3.24	95.55±2.62	**97.48±1.37**
	0.2	98.08±1.98	98.02±1.17	96.65±2.11	96.08±1.73	**98.24±1.56**
	0.3	98.13±2.01	98.38±1.23	97.30±1.65	96.23±1.48	**98.44±1.38**
Hdigt	0.01	96.53±1.69	95.83±2.28	96.11±0.69	93.16±0.38	**96.93±2.13**
	0.02	96.87±1.37	96.17±1.12	96.22±0.62	93.47±0.34	**98.02±1.19**
	0.03	97.19±1.19	96.49±0.93	96.27±0.56	93.74±0.31	**98.14±1.02**
VoxCeleb	0.01	66.03±0.84	65.18±1.63	69.24±2.04	66.09±1.93	**71.27±1.87**
	0.02	78.27±0.66	75.12±1.55	79.57±1.58	74.11±1.76	**82.67±1.64**
	0.03	82.40±0.68	83.71±1.59	86.01±1.78	83.83±1.48	**88.29±1.35**
Youtube	0.01	OM	OM	56.13±0.59	66.83±1.33	**73.02±0.94**
	0.02	OM	OM	63.38±0.56	72.99±1.15	**76.39±0.74**
	0.03	OM	OM	67.09±0.62	75.60±1.24	**79.01±0.82**



(a) Mfeat (b) ORL (c) MSRC-v1

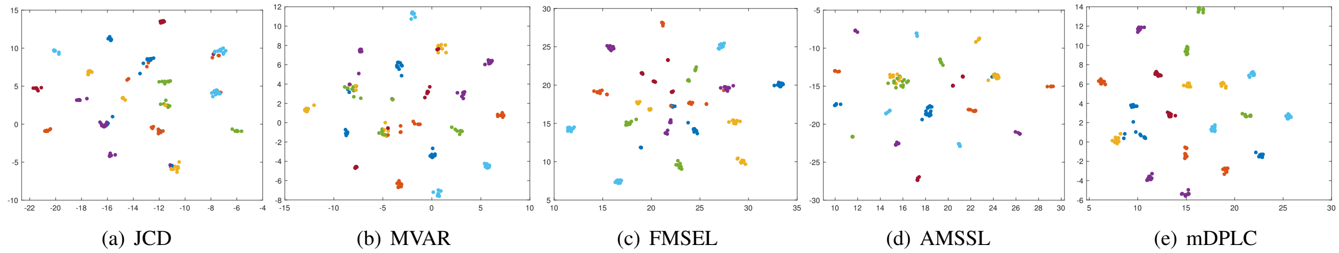
Fig. 4. The view weights with the number of iterations on multi-view datasets

of-memory error that occurred in training processes. It should be pointed out that LWGL, ERL, IMVGCN, and WFGSC) are transductive methods and cannot predict testing samples with their trained models or networks, thus we evaluate their performance using the results on unlabeled samples (reported in Table VII). In comparison to these state-of-the-art multi-view methods, the proposed mDLPC consistently demonstrates superior results on most datasets and achieves a significant performance improvement for both unlabeled samples and testing samples. This fully validates the effectiveness and superiority of mDLPC in leveraging different views simultaneously from both aspects of regression losses and similarity structures. Moreover, mDLPC equipped with the acceleration solution can efficiently tackle large-scale datasets, whereas some graph-based methods encounter the out-of-memory error (i.e., OM) on the YouTube dataset. Although the regression-based competitors (i.e., JCD and MVAR) and the



(a) ASL (b) ADLSR (c) FME (d) APGE (e) RER (f) SFAG (g) DLPC

Fig. 5. The visualization of the learned projection subspaces of different methods on the single-view Coil20 dataset.



(a) JCD (b) MVAR (c) FMSEL (d) AMSSL (e) mDPLC

Fig. 6. The visualization of the learned feature projection subspaces of different methods on the multi-view ORL dataset.

TABLE IX
THE RUNTIME (IN SECONDS) OF MULTI-VIEW METHODS ON 8 DATASETS

Datasets	FMSEL	AMSSL	LWGL	ERL	JCD	MVAR	IMVGCN	WFGSC	mDLPC
MSRC-v1	0.10	0.09	0.04	0.08	0.04	0.08	1.65	2.77	0.13
ORL	0.13	0.12	0.08	0.09	0.08	0.06	2.97	4.10	0.34
BDGP	3.40	3.19	2.38	3.09	0.34	1.09	68.44	100.29	2.92
Mfeat	3.23	1.65	1.78	2.38	0.10	0.55	71.03	120.01	2.18
Scene15	10.26	6.38	5.83	12.39	1.07	0.98	122.36	180.48	4.82
Hdigit	50.75	32.40	30.49	81.76	3.13	4.33	345.28	490.49	4.14
VoxCeleb	91.02	50.27	47.84	176.11	13.01	19.36	468.91	732.80	17.89
Youtube	OM	OM	OM	OM	22.59	53.98	509.12	1059.04	43.28

deep network-based competitors (i.e., IMVGCN and WFGSC) can be run on the large-scale datasets, their performance is inferior to mDLPC on all datasets.

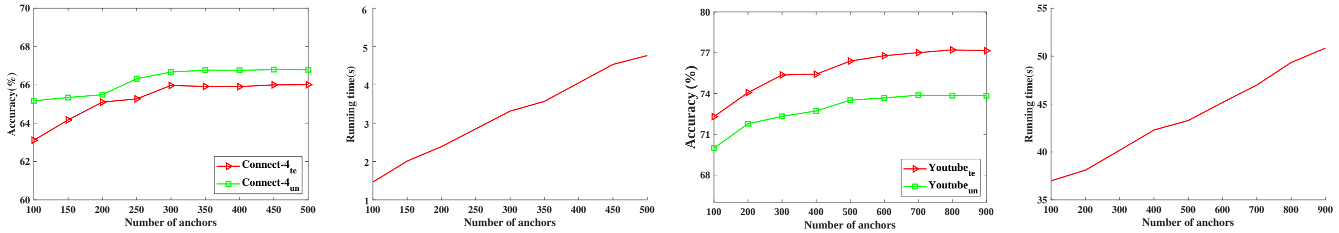
Table IX records the runtime on multi-view datasets, from which we can find that mDLPC likewise obtains highly competitive efficiency. For example, on the VoxCeleb dataset, mDLPC consumes the 17.89s runtime, while six competitors require more than 40s. On the YouTube dataset, the runtime of mDLPC merely accounts for about 10% and 5% of those in IMVGCN and WFGSC, respectively. These findings validate the scalability of the anchor-based acceleration strategy and position mDLPC as an effective semi-supervised learning framework for multi-view tasks. Meanwhile, mDLPC discriminates different views and fuses them from the aspects of regression losses and similarity structures. Fig. 4 depicts the evolution trend of the weights assigned to different views. We observe that although different views contribute variously to models, mDLPC can effectively balance them and assign a proper weight to each view, highlighting prominent views and simultaneously reducing the effects of poor views. Therefore, the proposed multi-view extension not only broadens the application scenarios of DLPC but also facilitates performance improvement via the effective fusion of multiple views.

C. Visualization

To intuitively display the learned feature projection subspace, the t-SNE [44] is used to map the projection subspace into a two-dimensional space. Fig. 5 and Fig. 6 show the results of different methods on the subsets of single-view Coil20 (400 samples from 20 classes) and multi-view ORL (200 samples from 20 classes), respectively. Specifically, Figs. 5(a)-(b) and Figs. 6(a)-(b) are the visualization results of regression-based methods, and Figs. 5(c)-(f) and Figs. 6(c)-(e) correspond to the results of graph-based methods. From Figs. 5 and 6, we note that regression-based methods can achieve compact subspaces by projecting samples to the independent class indicators, yet there still exist several samples that cannot be distinguished. The main reason is that the ground truth label information contained in the class indicators is not enough to separate all boundary samples without using the similarity structures to further enrich label information. Meanwhile, we find that there exist overlap areas between different classes in the projection subspaces of graph-based methods, which is attributed to the incorrect label propagation on the classification boundary. This further indicates that it is suboptimal for graph-based methods to directly leverage propagated labels as regression targets to guide the feature projection learning. Differing from these methods, the proposed DLPC achieves superior results, and the learned projection subspaces can identify almost all samples as depicted in Fig. 5(h) and Fig. 6(e). This demonstrates that the collaborative interaction of similarity structures and projected losses indeed facilitates the accurate propagation of label information, so as to enhance the discrimination of projection subspaces for different classes.

D. Effect of Anchor Points' Number

In the experiments of Section V-B, we use k-means clustering to generate anchor points to accelerate DLPC, and the numbers of anchors are recorded in Table III. To analyze the impact of anchors on the performance and scalability of the acceleration solution designed in Section IV-C, we conduct the quantitative experiments, in which the numbers of anchors vary in the range of {100, 150, · · · , 500} and {100, 200, · · · , 1000} on the Connect-4 and Youtube datasets, respectively. As the number of anchors increases, Fig. 7 shows the accuracy



(a) ACC versus the number of anchors (b) Time versus the number of anchors (c) ACC versus the number of anchors (d) Time versus the number of anchors

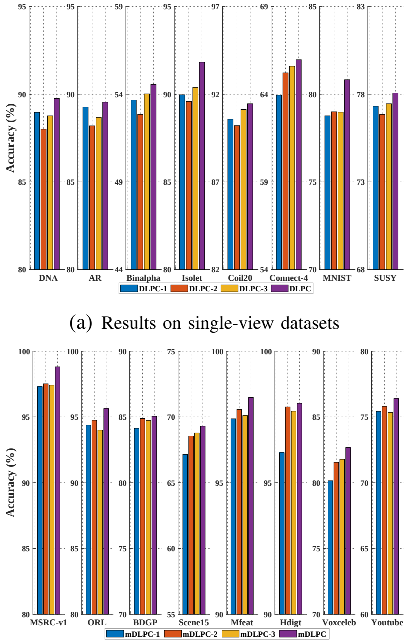
Fig. 7. The accuracy and runtime in relation to the number of anchors, in which (a) and (b) display the results on the single-view Connect-4 dataset, (c) and (d) illustrate the results on the multi-view YouTube dataset.

and the runtime of DLPC using 2% labeled samples. From Fig. 7, we note that there exist obviously different variation trends in the accuracy and runtime with the number of anchors. Specifically, when the number of anchors increases, the accuracy initially improves rapidly, and then it tends to be a stable value later on (as depicted in Figs. 7(a) and (c)). However, the running time in Figs. 7(b) and (d) keeps significantly increasing, indicating the degraded scalability of DLPC. This demonstrates that although increasing the number of anchors can enhance their ability to represent the original data (see the improved classification accuracy), it in turn involves more computational costs (see the significantly increased running time). Therefore, more anchors are beneficial to improve the performance (such as the sensitivity and robustness of the model to generated anchors), but they likewise significantly reduce the model scalability. This means that the number of anchors is not necessarily the more, the better, and we have to make a compromise between performance and scalability in practical tasks. Considering these factors, it is still necessary to conduct further research on how to determine the optimal number of anchors and thus learn more representative anchors.

E. Ablation Study

In this section, ablation experiments are conducted to investigate different components in the proposed DLPC.

Single-view DLPC. Three variants of single-view DLPC are designed: DLPC-1 directly uses the fixed class indicator (i.e., $\mathbf{t}_k$) as a regression target without assigning the nonnegative adjustment vector $\mathbf{e}_i^k$ to each class; DLPC-2 utilizes the prediction label $\mathbf{F}$ that is learned from the first and second terms of Eq. (15) to guide feature projection learning, which discards the interaction between similarity relationships and regression losses; DLPC-3 uses the initial graph $\mathbf{A}$ to replace the similarity graph learning model. Fig. 8(a) shows the classification results on testing samples with 20% or 2% labeled samples, from which we observe that DLPC outperforms all variants on all datasets. Specifically, the classification accuracies of DLPC-1 are considerably inferior to those of DLPC, which indicates that the nonnegative adjustment vector $\mathbf{e}_i^k$ is helpful to enhance the discriminative capability of regression losses, thereby correcting propagated labels and facilitating projection learning. Meanwhile, DLPC outperforms DLPC-2, which indicates the effectiveness of exploiting the similarity relations and regression losses collaboratively. Besides, DLPC also achieves higher accuracy than DLPC-3, showing that dynamically learning the similarity relations among samples can enhance the performance of the model.

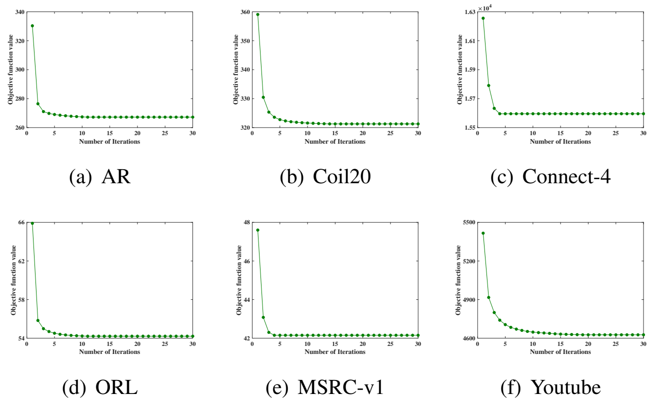


(a) Results on single-view datasets

(b) Results on multi-view datasets

Fig. 8. The comparison of DLPC and its three variants. (a) and (b) show the results on the single-view and multi-view datasets, respectively.

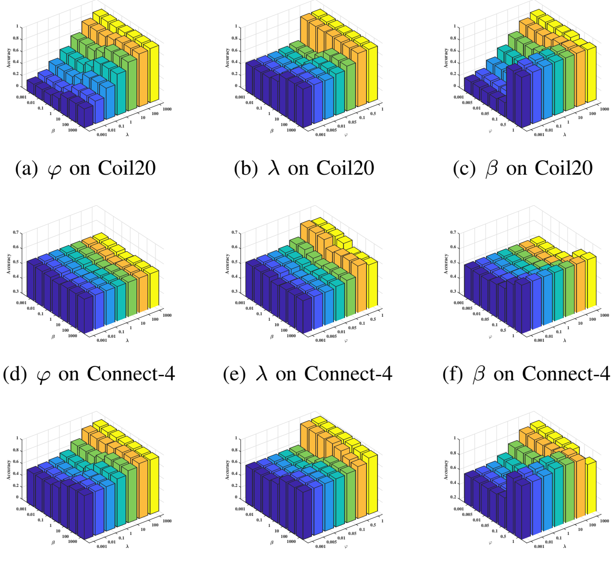
Multi-view DLPC. Three variants of the multi-view DLPC (i.e., mDLPC) are designed: mDLPC-1 uses the fixed view weights (i.e., $\alpha_v = 1/V$ for $v = 1, \cdots, V$); mDLPC-2 considers the difference of each view only from the aspect of similarity graph; mDLPC-3 distinguishes different views from the aspect of regression (projection) losses and replaces the adaptive graph fusion with a direct fusion manner (i.e., $\mathbf{S} = \sum_{v=1}^{V} \mathbf{A}^v / V$) during label propagation. As shown in Fig. 8 (b), mDLPC-2 and mDLPC-3 exhibit superior performance to mDLPC-1, which indicates that completely ignoring the distinctions among views indeed compromises the ultimate performance. Meanwhile, mDLPC consistently outperforms mDLPC-2 and mDLPC-3, highlighting the importance of distinguishing different views from diverse aspects (i.e., regression losses and similarity structures). Therefore, it is effective for the multi-view extension of DLPC to fully preserve and leverage the distinction and correlation among views from



(a) AR (b) Coil20 (c) Connect-4

(d) ORL (e) MSRC-v1 (f) Youtube

Fig. 9. Convergence curves of DLPC with 20% or 2% labeled samples.



(a) φ on Coil20 (b) λ on Coil20 (c) β on Coil20

(d) φ on Connect-4 (e) λ on Connect-4 (f) β on Connect-4

(g) φ on MSRC-v1 (h) λ on MSRC-v1 (i) β on MSRC-v1

Fig. 10. Accuracy variations of our model with varying φ, λ and β on Coil20, Connect-4 and MSRC-v1 datasets.

the aspects of both regression losses and similarity structures, positively facilitating label propagation and view-specific projection learning.

F. Convergence and Sensitivity Analysis

To verify the convergence of DLPC, Fig. 9 shows the convergence curves with 20% or 2% labeled samples on single-view and multi-view datasets, where the parameters λ, β, and φ are set to 1, 1, and 0.1, respectively. From Fig. 9, it can be observed that the objective function value sharply decreases in the first few iterations and converges quickly (in less than 10 iterations), demonstrating the fast convergence of the optimization strategy for DLPC.

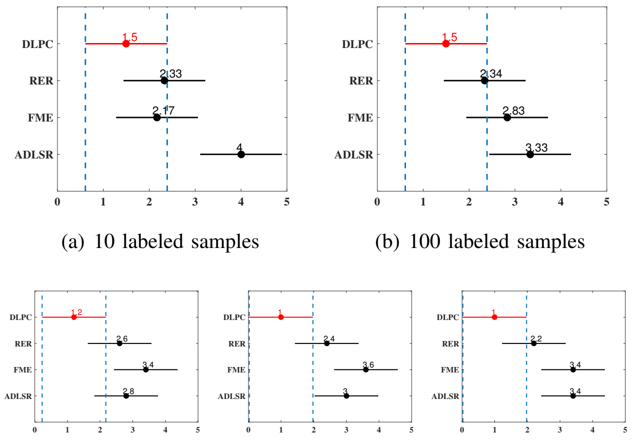
Considering that DLPC involves three parameters that need to be manually tuned, it is essential to investigate their effects. Specifically, λ balances the role of the feature projection learning, β controls the mismatch between the optimal graph $\mathbf{S}$ and the initial graph $\mathbf{A}$, and φ controls the use of regression losses. To this end, we conduct experiments on the Coil20, MSRC-v1, and Connect-4 datasets with 20% or 2% labeled samples, respectively. Fig. 10 illustrates the results with different values of parameters, showing that DLPC can achieve better performance as λ, β, and φ gradually increase. This further demonstrates the effectiveness of incorporating these components into DLPC for semi-supervised classification. In practical applications, we suggest fixing λ as 10 first and then determining parameters φ and β using the grid search.

G. Statistical Analysis and Discussion

To verify the superiority of the proposed DLPC over other methods with statistical significance, the Friedman test with the two-tailed Bonferroni-Dunn test [45] is further used to analyze the comprehensive performance of DLPC and other competitors. Specifically, the performance of DLPC and another competitor is significantly different, if their average ranks calculated according to their performance on multiple datasets differ by at least the critical difference (CD): $CD = q_{\varrho}\sqrt{\frac{p(p+1)}{6M}}$, where p is the number of methods, M is the number of datasets, ϱ denotes the significance level, and q_{ϱ} is the critical value. According to the above experiment results, we analyze the comprehensive difference between DLPC and other methods on single-view and multi-view datasets, respectively.

On the single-view datasets that include six benchmarks and five real-world datasets, three representative competitors, i.e., ADLSR, FME, and RER, are chosen to compare with DLPC. At a significance level of 0.05, we have $q_{0.05} = 2.39$, and $CD = 1.78$ on the benchmarks and $CD = 1.95$ on the real-world datasets, respectively. Fig. 11 shows the statistical significance results of DLPC and three competitors using different numbers/ratios of labeled samples in training processes. On the eight multi-view datasets, two representative competitors, i.e., JCD and AMSSL, are chosen to compare with mDLPC. We have $q_{0.05} = 2.24$, and $CD = 1.12$ with $p = 3$ and $M = 8$. Fig. 12 demonstrates the significance test results of mDLPC and other competitors on multi-view datasets. We can observe that the average rank differences between DLPC and ADLSR on the benchmarks (as shown in Figs. 11 (a) and (b)), the differences between DLPC and FME, and the differences between DLPC and ADLSR with 20% and 30% labeled ratios on the real-world datasets (as shown in Figs. 11 (c), (d), and (e)), are highly significant. This indicates that, in most cases, the proposed DLPC indeed significantly outperforms the typical regression-based method (i.e., ADLSR) and the typical graph-based method (i.e., FME). As shown in Fig. 12, we note that the differences between mDLPC and AMSSL, and the differences between mDLPC and JCD with 20% (or 2%) and 30% (or 3%) labeled ratios, are greater than CD, which means that mDLPC is significantly better than AMSSL and JCD on the multi-view datasets.

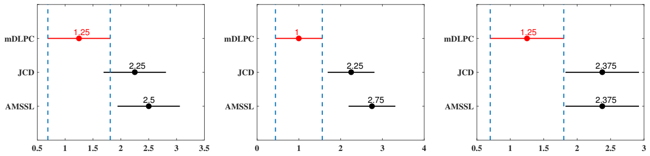
Generally, the statistical significance tests further validate the superiority of the proposed DLPC over other methods. We believe that the superior performance achieved by DLPC can be attributed to the following factors: (1) Based on the deep analyses of graph-based methods and regression-based methods, DLPC simultaneously leverages regression losses and similarity structures to guide the learning process, in which the



(a) 10 labeled samples (b) 100 labeled samples

(c) 10% labeled ratio (d) 20% labeled ratio (e) 30% labeled ratio

Fig. 11. The Friedman tests for the comprehensive performance of DLPC and other methods on single-view datasets, in which (a) and (b) denote the results on the benchmarks, and (c), (d), and (e) are the results (i.e., performance on testing samples) on the five real-world datasets. The dots denote the average ranks, the blue bars are the critical values at the 0.05 significance level, and the methods having non-overlapped bars are significantly inferior to DLPC.



(a) 10% (1%) labeled ratio (b) 20% (2%) labeled ratio (c) 30% (3%) labeled ratio

Fig. 12. The Friedman tests of mDLPC and others on multi-view datasets.

labels propagated via similarity graphs and the losses of projecting samples onto different classes can collaboratively interact with each other, effectively making up for the deficiencies of previous methods in terms of projection learning and label prediction; (2) Except for dynamically optimizing graph structures during propagating labels, DLPC assigns the independent class label associated with a nonnegative adjustment vector for each sample, not only enlarging the distance between different classes in projected spaces, but also making regression losses more discriminative for the classification boundary; (3) We have designed effective extension strategies for the proposed semi-supervised learning framework, enabling it to apply to large-scale and multi-view data scenarios. For example, the mDLPC fully leverages the complementarity and consistency among views from the aspects of both regression losses and similarity structures, facilitating label propagation and view-specific projection learning.

VI. Conclusion and Future Work

In this paper, we propose a scalable framework, called Discriminative Label Propagation and Correction (DLPC), to overcome the limitations of existing semi-supervised learning paradigms. Unlike previous methods that rely solely on similarity graphs or regression losses for training models, DLPC ensures more accurate prediction labels and discriminative projection subspaces through the collaborative interaction and dynamic learning of regression losses and similarity graphs. This manner can effectively reduce misclassifications of boundary samples. Furthermore, our multi-view DLPC adaptively distinguishes and integrates graph structures and projected losses across different views, thereby balancing their complementarity and correlation. To improve the solution efficiency, an anchor-based acceleration strategy has been developed for DLPC, thereby scaling DLPC to large-scale problems. Extensive experiments conducted on various datasets demonstrate the effectiveness and superiority of DLPC in both single-view and multi-view tasks.

Despite achieving advancements, the proposed semi-supervised learning framework still has the following limitations. Firstly, although the performance of DLPC was verified through empirical research and quantitative analyses, there is no rigorous theoretical analysis regarding the robustness and reliability of DLPC for boundary samples. Therefore, in future research, we aim to conduct a formal analysis from the perspectives of the generalization error bound [31] and the classification margin [46] to quantify the impact of samples with ambiguous labels or similarity structures. Secondly, the proposed DLPC employs k-means to generate anchor points, which completely separates the anchor generation from label propagation. This means that the pre-generated anchors are often not the optimal choice for label propagation. While increasing the number of anchors can maintain better accuracy, such a manner substantially compromises the scalability of DLPC. We plan to use the self-representation strategy (e.g., $\|\mathbf{X} - \mathbf{ZS}\|_F^2$) to dynamically update the anchor points and learn the similarity relationships (i.e., $\mathbf{S}$) during label propagation. These limitations and potential solutions will drive our continued research in the field of semi-supervised learning.

References

[1] G. Li, Z. Yu, K. Yang, C. P. Chen, and X. Li, “Ensemble-enhanced semi-supervised learning with optimized graph construction for high-dimensional data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 2, pp. 1103–1119, 2025.

[2] N. Sun, T. Luo, W. Zhuge, H. Tao, C. Hou, and D. Hu, “Semi-supervised learning with label proportion,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 877–890, 2023.

[3] G. Zeng, H. Peng, A. Li, J. Wu, C. Liu, and S. Y. Philip, “Scalable semi-supervised clustering via structural entropy with different constraints,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 1, pp. 478–492, 2025.

[4] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press Cambridge, 2006.

[5] D. Wang, F. Nie, and H. Huang, “Large-scale adaptive semi-supervised learning via unified inductive and transductive model,” in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 482–491.

[6] M. Luo, L. Zhang, F. Nie, X. Chang, B. Qian, and Q. Zheng, “Adaptive semi-supervised learning with discriminative least squares regression.” in *International Joint Conference on Artificial Intelligence*, 2017, pp. 2421–2427.

[7] L. Zhang, M. Luo, Z. Li, F. Nie, H. Zhang, J. Liu, and Q. Zheng, “Large-scale robust semisupervised classification,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 907–917, 2018.

[8] Z. Song, X. Yang, Z. Xu, and I. King, “Graph-based semi-supervised learning: A comprehensive review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8174–8194, 2023.

[9] H. Nie, Q. Li, Z. Wang, H. Zhao, and F. Nie, “Semisupervised subspace learning with adaptive pairwise graph embedding,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 18 105–18 119, 2024.

[10] R. Sheikhpour, F. Saberi-Movahed, M. Jalili, and K. Berahmand, “Semi-supervised feature selection with concept factorization and robust label learning,” *Pattern Recognition*, p. 112317, 2026.

[11] Y. Pi, Y. Wu, Y. Huang, Y. Shi, and S. Wang, “Inhomogeneous diffusion-induced network for multiview semi-supervised classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 5, pp. 8606–8618, 2025.

[12] X. Jia, X. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, and D. Yue, “Semi-supervised multi-view deep discriminant representation learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2496–2509, 2021.

[13] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, “Triplet adaptation framework for robust semi-supervised learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 8056–8073, 2024.

[14] L. Zhang, R. Song, W. Tan, L. Ma, and W. Zhang, “Igcn: A provably informative gcn embedding for semi-supervised learning with extremely limited labels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 8396–8409, 2024.

[15] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[16] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang, “Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction,” *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1921–1932, 2010.

[17] H. Nie, Q. Wu, H. Zhao, W. Ding, and M. Deveci, “Flexible adaptive graph embedding for semi-supervised dimension reduction,” *Information Fusion*, p. 101872, 2023.

[18] J. Bao, M. Kudo, K. Kimura, and L. Sun, “Robust embedding regression for semi-supervised learning,” *Pattern Recognition*, vol. 145, p. 109894, 2024.

[19] X. Qi, H. Zhang, and F. Nie, “Discriminative semi-supervised feature selection via a class-credible pseudo-label learning framework,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2024, pp. 6895–6899.

[20] M. Mohammadi, K. Berahmand, S. Azizi, R. Sheikhpour, and H. Khosravi, “Semi-supervised adaptive symmetric nonnegative matrix factorization for multi-view clustering,” *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 6, pp. 4967–4981, 2025.

[21] J. Bi, F. Dornaika, and J. Charafeddine, “Linear projection fused graph-based semi-supervised learning on multi-view data,” *Artificial Intelligence Review*, vol. 58, no. 10, p. 309, 2025.

[22] Z. Li, Q. Qiang, B. Zhang, F. Wang, and F. Nie, “Flexible multi-view semi-supervised learning with unified graph,” *Neural Networks*, vol. 142, pp. 92–104, 2021.

[23] S. Bahrami, F. Dornaika, and A. Bosaghzadeh, “Joint auto-weighted graph fusion and scalable semi-supervised learning,” *Information Fusion*, vol. 66, pp. 213–228, 2021.

[24] N. Liang, Z. Yang, J. Chen, Z. Li, and S. Xie, “Label-weighted graph-based learning for semi-supervised classification under label noise,” *IEEE Transactions on Big Data*, vol. 10, no. 1, pp. 1547–65, 2024.

[25] C. Zhang, X. Zhu, Z. Wang, Y. Zhong, W. Sheng, W. Ding, and B. Jiang, “Discriminative multi-view fusion via adaptive regression,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 6, pp. 3821–3833, 2024.

[26] H. Tao, C. Hou, F. Nie, J. Zhu, and D. Yi, “Scalable multi-view semi-supervised classification via adaptive regression,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4283–4296, 2017.

[27] A. Huang, Z. Wang, Y. Zheng, T. Zhao, and C.-W. Lin, “Embedding regularizer learning for multi-view semi-supervised classification,” *IEEE Transactions on Image Processing*, vol. 30, pp. 6997–7011, 2021.

[28] S. Wang, Z. Chen, S. Du, and Z. Lin, “Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5042–5055, 2022.

[29] Z. Wu, X. Lin, Z. Lin, Z. Chen, Y. Bai, and S. Wang, “Interpretable graph convolutional network for multi-view semi-supervised learning,” *IEEE Transactions on Multimedia*, vol. 25, pp. 8593–8606, 2023.

[30] J. Bi and F. Dornaika, “Sample-weighted fused graph-based semi-supervised learning on multi-view data,” *Information Fusion*, vol. 104, p. 102175, 2024.

[31] B. Jiang, H. Chen, B. Yuan, and X. Yao, “Scalable graph-based semi-supervised learning through sparse bayesian model,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2758–2771, 2017.

[32] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 977–986.

[33] C. Xu, J. Si, Z. Guan, W. Zhao, Y. Wu, and X. Gao, “Reliable conflictive multi-view learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16 129–16 137, 2024.

[34] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[35] M. Li, S. Wang, X. Liu, and S. Liu, “Parameter-free and scalable incomplete multiview clustering with prototype graph,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 300–310, 2024.

[36] S. Wang, X. Liu, S. Liu, J. Jin, W. Tu, X. Zhu, and E. Zhu, “Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5882–5895, 2022.

[37] M. Noordewier, G. Towell, and J. Shavlik, “Training knowledge-based neural networks to recognize genes in dna sequences,” *Advances in Neural Information Processing Systems*, vol. 3, 1990.

[38] B. Li, J. Wang, Z. Yang, J. Yi, and F. Nie, “Fast semi-supervised self-training algorithm based on data editing,” *Information Sciences*, vol. 626, pp. 293–314, 2023.

[39] W. Zhuge, C. Hou, S. Peng, and D. Yi, “Joint consensus and diversity for multi-view semi-supervised classification,” *Machine Learning*, vol. 109, pp. 445–465, 2020.

[40] B. Jiang, J. Liu, Z. Wang, C. Zhang, J. Yang, Y. Wang, W. Sheng, and W. Ding, “Semi-supervised multi-view feature selection with adaptive similarity fusion and learning,” *Pattern Recognition*, vol. 159, p. 111159, 2025.

[41] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, “Partially view-aligned representation learning with noise-robust contrastive loss,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 1134–1143.

[42] S. Qiu, F. Nie, X. Xu, C. Qing, and D. Xu, “Accelerating flexible manifold embedding for scalable semi-supervised learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2786–2795, 2018.

[43] Z. Ibrahim, A. Bosaghzadeh, and F. Dornaika, “Joint graph and reduced flexible manifold embedding for scalable semi-supervised learning,” *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9471–9495, 2023.

[44] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[45] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[46] Y. Li, S. Wang, and Z. Zhou, “Graph quality judgement: A large margin expedition,” in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 1725–1731.

Bingbing Jiang (Member, IEEE) received the B.Sc. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2014, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2019. He is currently an Associate Professor at Hangzhou Normal University, Hangzhou. 400065, China. His research interests include semi-supervised learning, multi-view learning, and feature selection. He serves as an Editorial Board member of Applied Soft Computing and an Associate Editor of Neurocomputing.

Jie Wen (Senior Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Shenzhen, China, in 2019. He is currently an Associate Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interests include image and video enhancement, pattern recognition, and machine learning. He serves as an Associate Editor/Area Editor/Editorial Board member of IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Circuits and Systems for Video Technology, Pattern Recognition, Information Fusion, CAAI Transactions on Intelligence Technology, et al.

Zidong Wang (Fellow, IEEE) received the B.Sc. degree in mathematics from Suzhou University, Suzhou, China, in 1986, and the M.Sc. degree in applied mathematics and the Ph.D. degree in electrical engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1990 and 1994, respectively. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany, and U.K. He is currently a Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, Uxbridge, U.K. He has authored or coauthored more than 700 papers in international journals. His research interests include dynamical systems, signal processing, bioinformatics, and control theory and applications.

Dr. Wang was a recipient of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, and William Mong Visiting Research Fellowship of Hong Kong. He is (or was) the Editor-in-Chief for International Journal of Systems Science, the Editor-in-Chief for Neurocomputing, an Editor-in-Chief for Systems Science & Control Engineering, and an Associate Editor for 12 international journals, including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics-Part C. He is a Member of the Academia Europaea, a Member of the European Academy of Sciences and Arts, an Academician of the International Academy for Systems and Cybernetic Sciences, a Fellow of the Royal Statistical Society, and a Member of the program committee for many international conferences.

Weiguo Sheng (Member, IEEE) received the M.Sc. degree in information technology from the University of Nottingham, UK, in 2002, and a Ph.D. degree in computer science from Brunel University London, UK, in 2005. He is currently a Professor at Hangzhou Normal University, Hangzhou, China. His research interests include evolutionary computation and pattern recognition. He is an Associate Editor of IEEE Transactions on Emerging Topics in Computational Intelligence.

Zhiwen Yu (Senior Member, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008. He is a professor with the School of Computer Science and Engineering, South China University of Technology, China. He is an Associate Editor of IEEE Transactions on Systems, Man, and Cybernetics: Systems.

Huanhuan Chen (Fellow, IEEE) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2004 and the Ph.D. degree from the University of Birmingham, Birmingham, UK, in 2008. He is currently a Full Professor in the School of Computer Science and Technology, USTC. His research interests include neural networks and machine learning. He served as an Associate Editor of Knowledge-Based Systems.

Weiping Ding (Senior Member, IEEE) received the Ph.D. degree in Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. He is a Full Professor with the School of Artificial Intelligence and Computer Science, Nantong University, Nantong, China. His main research directions involve deep neural networks, granular data mining, and machine learning. He ranked within the top 2% Ranking of Scientists in the World by Stanford University (2020-2024). He has published over 360 articles, including over 170 IEEE Transactions papers. He serves as an Associate Editor/Area Editor/Editorial Board member of more than 10 prestigious international journals, such as IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Fuzzy Systems, IEEE/CAA Journal of Automatica Sinica, IEEE Transactions on Emerging Topics in Computational Intelligence, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Intelligent Vehicles, IEEE Transactions on Artificial Intelligence, Information Fusion, Information Sciences, Neurocomputing, Applied Soft Computing, Engineering Applications of Artificial Intelligence, Swarm and Evolutionary Computation, et al.