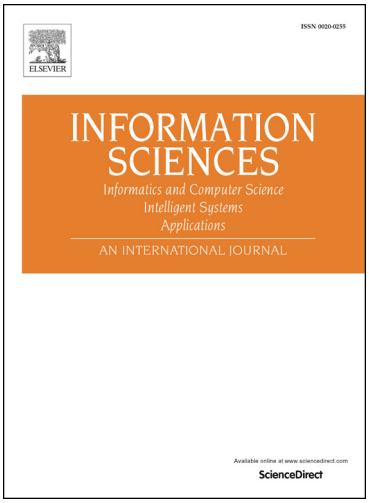
Journal Pre-proof



Geometric Localized Graph Convolutional Network for Multi-view Semi-supervised Classification

Aiping Huang, Jielong Lu, Zhihao Wu, Zhaoliang Chen, Yuhong Chen et al.

PII: S0020-0255(24)00683-2
DOI: https://doi.org/10.1016/j.ins.2024.120769
Reference: INS 120769

To appear in: *Information Sciences*

Received date: 10 July 2023
Revised date: 14 December 2023
Accepted date: 21 May 2024

Please cite this article as: A. Huang, J. Lu, Z. Wu et al., Geometric Localized Graph Convolutional Network for Multi-view Semi-supervised Classification, *Information Sciences*, 120769, doi: https://doi.org/10.1016/j.ins.2024.120769.

Highlights

- Propose an end-to-end framework for multi-view semi-supervised classification.
- Utilize diffusion map to obtain the geometric structure of the feature space of each view.
- Propose a truncated diffusion correlation function to obtain a reliable sparse graph.

Geometric Localized Graph Convolutional Network for Multi-view Semi-supervised Classification

Aiping Huang[a], Jielong Lu[b,c], Zhihao Wu[b,c], Zhaoliang Chen[b,c], Yuhong Chen[b,c], Shiping Wang[b,c], Hehong Zhang[b,c,*]

[a]*School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 350108, China*
[b]*College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China*
[c]*Key Laboratory of Intelligent Metro, Fujian Province University, Fuzhou 350108, China*

Abstract

Multi-view learning has received increasing attention in recent years due to its ability to leverage valuable patterns hidden in heterogeneous data sources. While existing studies have achieved encouraging results, especially those based on graph convolutional networks, they are still limited in their ability to fully exploit the connectivity relationships between samples and are susceptible to noise. To address the aforementioned limitation, we propose a framework called geometric localized graph convolutional network for multi-view semi-supervised classification. This framework utilizes a diffusion map to obtain the geometric structure of the feature space of multiple views and constructs a stable distance matrix that considers the local connectivity of nodes on the geometric structure. Additionally, we propose a truncated diffusion correlation function that maps the distance matrix of each view into correlations between samples to obtain a reliable sparse graph. To fuse the features, we use learnable weights to concatenate the coordinates of the geometric structure. Finally, we obtain a graph embedding of the fused feature and topology by using graph convolutional networks. Comprehensive experiments demonstrate the superiority of the proposed method over other state-of-the-art methods.

Keywords: Multi-view learning, semi-supervised classification, diffusion map, manifold learning, graph convolution networks.

*Corresponding author.
Email addresses: sxxhap@163.com (Aiping Huang), jielonglu2022@163.com (Jielong Lu), zhihaowu1999@gmail.com (Zhihao Wu), chenzl123@outlook.com (Zhaoliang Chen), yhchen2320@163.com (Yuhong Chen), shipingwangphd@163.com (Shiping Wang), hzhang030@e.ntu.edu.sg (Hehong Zhang)

1. Introduction

With the continuous development of big data and models, information in the real world often comes from various information extractors. For instance, an object can be perceived through different human senses such as sight, touch, and smell. This diverse sensory input forms multi-view data. The objective of multi-view learning is that amalgamate information from various perspectives in order to enhance practical applications such as computer vision [1] [2] [3], node clustering [4] [5] [6], and machine learning [7] [8]. Typically, multi-view learning algorithms can achieve satisfactory performance when a sufficient number of labeled samples are obtained. However, in real-life scenarios, sample labeling can be constrained by labor and cost, especially for multi-view data. Therefore, multi-view semi-supervised classification is a more practical branch that leverages a small amount of labeled data to guide the prediction of a large amount of unlabeled data.

Graph-based multi-view semi-supervised classification algorithms have gained significant attention due to their ability to approximate the manifold structure of the samples [9]. These algorithms primarily rely on techniques such as random walk [10] [11], matrix decomposition [12] [13], and graph convolutional networks (GCNs) [5] [14] to obtain low-dimensional embeddings for downstream tasks. Among these methods, GCNs [15] have emerged as a powerful tool to extract more intricate semantic information from the feature space and have demonstrated success in various scenarios, including graph classification [16] [17] [18], link prediction [19] [20] [21], and recommendation systems [22] [23] [24]. Although existing GCN-based methods have achieved promising results on multi-view data, they continue to encounter the following challenges: 1) Directly calculating similarity using data from the original space may be affected by noise, resulting in suboptimal results [25] [26]; 2) Ignoring the geometric structure hidden in the feature spaces may lead to underutilization of the connections between samples [27]; 3) Assigning a fixed number of neighbors to each node may produce incorrect connections [28]. To address the challenges mentioned above, we introduce an effective framework termed Geometric Localized Graph Convolutional Network for Multi-view Semi-supervised Classification (GLGCN), which leverages diffusion maps to capture the geometric structure of the original features from different views. This allows us to construct a more robust distance metric that is less sensitive to noisy data, resulting in a more reliable topological connection among samples. To be more clear, we calculate the state transition matrix for the feature matrices of each view at a given scale. Then, we utilize eigenvalue decomposition to obtain the diffusion coordinates. To enable feature propagation across multiple views, we employ both feature fusion and topology fusion. Feature fusion involves assigning learnable weights to the diffusion coordinates of each view and

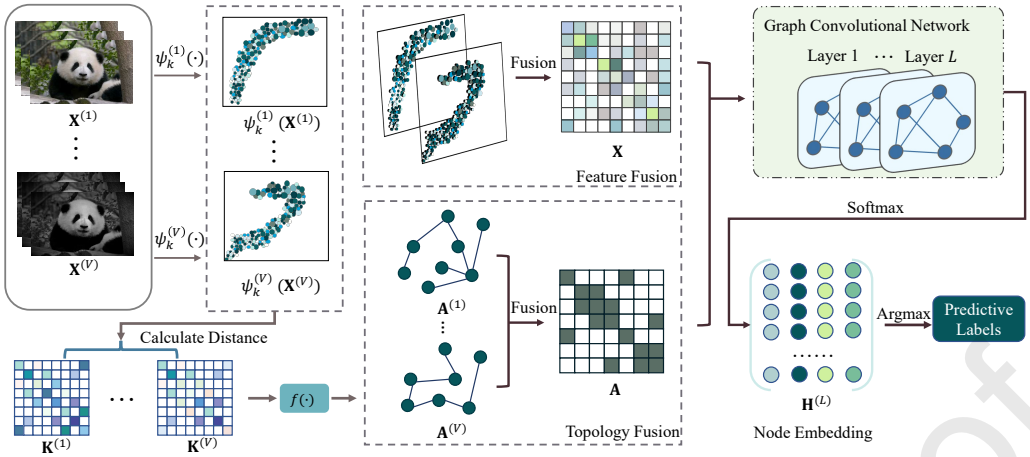


Figure 1: The proposed framework is founded on the use of diffusion maps to obtain robust distance metrics. Specifically, it performs a scale-specific diffusion map of the original space to extract the underlying geometric structure. Using this structure, a truncated diffusion similarity function is employed to obtain sparse topological connectivity. Finally, a graph convolutional network is utilized to integrate the fused graphs and features.

then concatenating them together. Topology fusion is achieved by first converting the diffusion distance matrix into a reliable and sparse adjacency matrix using a truncated diffusion correlation function.

The architecture of GLGCN is illustrated in Figure 1. The main contributions of this paper are summarized in the following aspects:

1) The proposed learning framework on graph diffusion fusion projects the original features onto a low-dimensional space and utilizes the diffusion distance to construct more robust connections between samples.

2) To obtain a more reliable and sparse topology, we propose a truncated diffusion correlation function, in which a suitable threshold is chosen on the distance between samples of each view.

3) The proposed method shows superior performance compared to other state-of-the-art multi-view semi-supervised methods.

This paper is organized as follows. In Section 2, we review related work on multi-view learning, graph convolutional networks, and multi-view dimensionality reduction. Section 3 elaborates on the proposed GLGCN method. Section 4 presents experimental results, and Section 5 concludes the work.

2. Related Work

In this section, we will review related work in multi-view learning, graph convolutional networks, and multi-view dimensionality reduction.

2.1. Muti-view Learning

Multi-view learning aims to improve the performance of machine learning tasks in different scenarios by extracting consistency and complementarity among views. Many studies have addressed multi-view data analysis. Canonical correlation analysis (CCA) and its kernel version are popular examples. Both methods aim to maximize the correlation between two views for consistent embeddings. The kernel version also adapts CCA to non-linear conditions for more complex real-world applications. Different from CCA, Zhao et al. [29] used non-negative matrix decomposition to obtain a hierarchical representation from multi-view data. Since deep neural networks possess strong non-linearities, Chen et al. [26] used a sparse autoencoder and a minimal threshold shrinkage function to train a network that considered both feature and topological fusions. Wen et al. [30] leveraged local geometric information and the unbalanced discriminative power of incomplete multi-view observations to obtain an effective incomplete multi-view clustering framework. Jia et al. [31] reduced the redundancy of learned representations by combining orthogonality and adversarial similarity constraints. All of these works illustrate that multi-view learning has greater potential than single-view learning.

2.2. Graph Convolutional Networks

The spectral convolution [32] was defined in the Fourier domain by the eigenmatrix obtained from the eigenvalue decomposition of the graph Laplacian matrix. Defferrard et.al [33] used Chebyshev's formula for approximation, eliminating the need to compute the eigenvectors of the Laplacian matrix. GCN [15] was built by restricting the Chebyshev polynomials to the first-order truncation as a way to alleviate the problem of overfitting the model in the local structure. Due to the powerful performance of GCN, many variants of it have been proposed and applied to different fields. Johannes et al. [34] used generalized graph diffusion to remove the restriction of using only first-order neighbors, alleviating the problem caused by the noise in the real graph. Li et al. [25] introduced GCN into multi-view learning by combining Laplace operators. You et al. [35] decomposed feature aggregation and feature transformation in the GCN training process to improve the learning speed of the model. Yang et al. [36] clustered spatially relevant features into several region-aware graphs and then explored the interconnections between regions using GCN. Wu et al. [37] proposed a semi-supervised multi-view convolutional network for webpage classification, featuring optimal graph structure

learning for individual views and integration of multi-view representations using an inter-view attention scheme. Jiang et al. [38] suggested coalescing different views by simultaneously fusing multiple feature projections, similarity maps, and adaptive weighting to fully preserve the correlation and differentiation among views. Despite the promising performance of these GCN-based approaches, there remains a shortage of exploration in establishing stable topologies for multi-view data which lacks nature topology.

2.3. Multi-view Dimensionality Reduction

Multi-view dimensionality reduction aims to efficiently utilize the existing data from multiple views to extract a coherent low-dimensional representation of multi-view data. Zhang et al. [39] proposed a method that used a kernel matching mechanism based on the Hilbert-Schmidt independence criterion to jointly maximize the correlation between different views. This allowed them to map high-dimensional spaces onto low-dimensional subspaces in a coherent manner. Yuan et al. [40] enforced predictiveness of the latent space by adaptively combining the correlations between the latent space and feature space, and maximizing the correlations between them. Chen et al. [26] obtained a low-dimensional representation of the data using a sparse autoencoder, which led to a more robust feature fusion. Wu et al. [28] aimed to obtain an efficient and consistent low-dimensional embedding across perspectives by incorporating orthogonal constraints into the optimization objective. By doing so, they constructed an effective multi-view network that could provide insights into the relationships between different views of the data. These multi-view methods of dimensionality reduction exploit the essential information in the data, reducing the amount of computation required by the model.

3. The Proposed Method

In this section, we elaborate on the problem formulation and optimization methodology of diffusion map fusion-based multi-view distance metric learning. In order to ease the presentation, commonly used notations are given in Table 1.

3.1. Theoretic Motivation

Given a set of data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^D$ for any $i \in [N]$, we can evaluate a kernel matrix $\mathbf{K} = [\mathcal{K}_{ij}]_{N\times N}$ on the sample space $\mathcal{X}$ using some kernel function, such as Gaussian kernel with $\mathbf{K}_{ij} = \exp\left(\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{\sigma^2}\right)$ where σ is a scale parameter for the kernel width. Accordingly, the diffusion-induced similarity matrix is given as $\mathbf{S}^{(\gamma)} = \mathbf{D}^{-\gamma}\mathbf{K}\mathbf{D}^{-\gamma}$ with $\mathbf{D} = [\mathbf{D}_{ij}]_{N\times N}$ as $\mathbf{D}_{ii} = \sum_{j=1}^N \mathbf{K}_{ij}$ for any $i \in [N]$. Consequently, we then construct a discrete-time Markov chain

Table 1: Commonly used notations with their descriptions.

Notations	Descriptions
$\{\mathbf{X}^{(v)} \in \mathbb{R}^{N\times D_v}\}_{v=1}^{V}$	Data with V views, N samples and D_v features
$\mathbf{Y}$, $\widehat{\mathbf{Y}}$	Real label and predictive label of multi-view data
$\mathbf{X} = [\mathbf{X}^{(1)} \parallel \cdots \parallel \mathbf{X}^{(V)}]$	Data matrix of multi-view feature concatenation
$\mathbf{K}$, $\mathbf{P}$, $\mathbf{D}$	Kernel matrix, transition matrix, degree matrix
$\mathcal{D}_k(\cdot,\cdot)$, $\boldsymbol{\psi}^k(\cdot)$	Diffusion distance, diffusion map at time step k
k, σ, τ, δ	Hyperparameters in the proposed model
$\{\boldsymbol{\Theta}^{(l)}\}_{l=1}^{L}$, $\boldsymbol{\alpha}$	Learnable parameters in the network
$\Phi(\cdot)$	Activation function in the network

$\mathbb{M}$ on $\mathcal{X}$ with the transition probability matrix $\mathbf{P}^{(\alpha)}$ (simply called $\mathbf{P}$) as the normalized kernel matrix $\mathbf{P}^{(\gamma)} = \mathbf{D}^{-1}\mathbf{S}^{(\gamma)}$ where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{S}_{ij}^{(\gamma)}$ for any $i \in [N]$. Here, $p(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{P}_{ij}^{(\gamma)}$ represents the one-step probability from state $\mathbf{x}_i$ to state $\mathbf{x}_j$, while $\mathbf{P}^k$ gives the k-step transition matrix. Generally, denote $p(\mathbf{x}_j, k|\mathbf{x}_i)$ as the probability from $\mathbf{x}_i$ to $\mathbf{x}_j$ within k steps. Accordingly, the diffusion distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ at time step k is defined as

$$\mathcal{D}_k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\mathbf{x}} \frac{(p(\mathbf{x}, k|\mathbf{x}_i) - p(\mathbf{x}, k|\mathbf{x}_j))^2}{\phi_0(\mathbf{x})}, \tag{1}$$

where $\phi_0(\mathbf{x})$ is the stationary distribution of the Markov chain $\mathbb{M}$ given by the eigenvector of the largest eigenvalue of $\mathbf{P}^{(\gamma)}$. The diffusion distance is calculated based on the local information of each node, meaning that each node only considers the information of its neighboring nodes. Therefore, the effect of changes in some nodes or edges on other nodes is limited, and this localized smoothness makes the calculation of the diffusion distance less affected by local perturbation on the graph.

Assume that $\mathbf{P}^{(\gamma)}$ has N eigenvalues $\{\lambda_i\}_{i=1}^{N}$ in descending order and the corresponding eigenvectors $\{\psi_i\}_{i=1}^{N}$, then the diffusion distance is equivalent to

$$\mathcal{D}_k^2(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l} \lambda_l^{2k} \left(\psi_l(\mathbf{x}_i) - \psi_l(\mathbf{x}_j)\right)^2 = \left\|\boldsymbol{\psi}^k(\mathbf{x}_i) - \boldsymbol{\psi}^k(\mathbf{x}_j)\right\|_2^2, \tag{2}$$

where $\boldsymbol{\psi}^k(\mathbf{x}) = [\lambda_1^k\psi_1(\mathbf{x}), \cdots, \lambda_d^k\psi_d(\mathbf{x})] \in \mathbb{R}^d$ is the diffusion map with $d = \max\{l \in \mathbb{N}, |\lambda_l|^k > \delta\,|\lambda_1|^k\}$ for a low-dimensional representation, where δ is preset to a value within the interval $(0, 1)$.

3.2. Problem Formulation

Given a multi-view dataset $\{\mathbf{X}^{(v)} \in \mathbb{R}^{N\times D_v}\}_{v=1}^{V}$ of n labels $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^{n}$ with $n << N$, the diffusion map time step k, diffusion map scale σ and hyperparameter

τ and δ, we can obtain the kernel matrices $\{\mathbf{K}^{(v)}\}_{v=1}^{V}$ for different views. We then normalize these kernel matrices to obtain a set of transition matrices $\{\mathbf{P}_{(v)}\}_{v=1}^{V}$ of multi-view data. Consequently, we can construct the diffusion distance $\mathcal{D}_k^{(v)}$ and diffusion map $\boldsymbol{\psi}_k^{(v)}$ of the v-view data $\mathbf{X}^{(v)}$. The proposed model attempts to address the following two considerations:

1) By treating data from different views differently and using alternative map scales, we can obtain more accurate distance metrics.

2) By measuring the distance distribution of node pairs across the sample space, we avoid assigning neighborhoods to pairs of nodes that are far apart.

Towards the goal of taking into account the considerations, the multi-view learning problem for semi-supervised classification is written as training an L-layer graph convolutional network,

$$\mathbf{H}^{(l)} = \sigma\left(\widehat{\mathbf{A}}\mathbf{H}^{(l-1)}\boldsymbol{\Theta}^{(l)}\right), \tag{3}$$

where $\widehat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\left[\sum_{v=1}^{V}\mathbf{A}^{(v)}\right]\mathbf{D}^{-\frac{1}{2}}$ is a normalized diffusion affinity matrix, where the adjacency matrix $\mathbf{A}^{(v)}$ is defined as $\mathbf{A}^{(v)} = f^{(v)}(\mathcal{D}_k^{(v)})$ and the diagonal matrix $\mathbf{D}$ is defined as $\mathbf{D}_{ii} = \sum_{j=1}^{N}[\sum_{v=1}^{V}\mathbf{A}^{(v)}]_{ij}$ for any $i \in [N]$. Here, the original input is the concatenation of multi-view diffusion maps $\mathbf{H}^{(0)} \doteq \mathbf{X} = \alpha_1\boldsymbol{\psi}_k^{(1)}(\mathbf{X}^{(1)}) \parallel \cdots \parallel \alpha_V\boldsymbol{\psi}_k^{(V)}(\mathbf{X}^{V})$ where $\boldsymbol{\psi}_k^{(v)}(\mathbf{X}^{(v)}) \in \mathbb{R}^{N\times D}$ is the diffusion map from the v-th view data at time step k, and $\boldsymbol{\alpha} = [\alpha_1; \cdots; \alpha_V]$ is a learnable weight vector constrained in $\mathcal{S}_{\boldsymbol{\alpha}} = \{\boldsymbol{\alpha} \in \mathbb{R}^V | \boldsymbol{\alpha}^\top\mathbf{1} = 1, \boldsymbol{\alpha} \geq \mathbf{0}\}$. The truncated diffusion correlation function for the v-th view is denoted as $f^v(\cdot)$, and we require it to be equipped with the following properties: First, it should be strictly monotonically decreasing and bounded, mapping the distance to the range $[0, 1]$; Second, no connection relation between sample pairs outside the given distance range should be constructed. To satisfy the above considerations, the truncated diffusion correlation function for v-th view can be written as

$$f(\mathcal{D}_k^{(v)}) = \text{ReLU}(1 - \tau\mathcal{D}_k^{(v)}), \tag{4}$$

where $\tau \in (0, 1)$ is a hyperparameter and ReLU$(\cdot)$ is a nonlinear activation function. The GCN forward propagation equation is obtained by bringing Equation 4 into Equation 3 as

$$\mathbf{H}^{(l)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\left[\sum_{v=1}^{V}\text{ReLU}(1 - \tau\mathcal{D}_k^{(v)})\right]\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l-1)}\boldsymbol{\Theta}^{(l)}\right). \tag{5}$$

3.3. Optimization Methodology

In this subsection, since the diffusion map fusion and the feature fusion can be computed and saved in advance, we only need to optimize the cross-entropy loss to

obtain the final graph representation. Accordingly, the output layer $\mathbf{H}^{(L)} \in \mathbb{R}^{N\times c}$ serves as the predicted label matrix $\widehat{\mathbf{Y}}$. Naturally, it is expected to have a minimum difference between the real label $\mathbf{Y}$ and the predicted label $\widehat{\mathbf{Y}}$, evaluated by their cross-entropy loss

$$\mathcal{L}_{cla} = -\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbf{Y}_{ij}\ln(\mathbf{H}_{ij}^{(L)}), \tag{6}$$

where n is the number of labeled samples used for training. The procedure for GLGCN is outlined in Algorithm 1.

Algorithm 1 Geometric Localized Graph Convolutional Network for Multi-View Semi-Supervised Classification (GLGCN)

Require: Multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$ with labels $\{\mathbf{y}_i\}_{i=1}^{n}$, diffusion map scale σ, time step k, hyperparameters τ and δ, network layer number L.
Ensure: Predictive labels $\{\widehat{\mathbf{y}}_i\}_{i=n+1}^{N}$.
$\triangleright$ Obtain transition matrices, diffusion distances and diffusion maps $\triangleleft$
for $v = 1 \to V$ **do**
  Calculate the kernel matrix $\mathbf{K}^{(v)}$ of the v-th view data;
  Compute the transition matrix $\mathbf{P}^{(v)}$ of the v-th view data using $\mathbf{K}^{(v)}$;
  Construct diffusion distance $\mathcal{D}_k^{(v)}$ and diffusion map $\boldsymbol{\psi}_k^{(v)}(\mathbf{X}^{(v)})$;
end for
Initialize graph convolution as $\widehat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\left[\sum_{v=1}^{V}\mathrm{ReLU}(1-\tau\mathcal{D}_k^{(v)})\right]\mathbf{D}^{-\frac{1}{2}}$, and input data as $\mathbf{H}^{(0)} = [\alpha_1\boldsymbol{\psi}_k^{(1)}(\mathbf{X}^{(1)}) \parallel \cdots \parallel \alpha_V\boldsymbol{\psi}_k^{(V)}(\mathbf{X}^{V})]$;
$\triangleright$ Label propagation $\triangleleft$
Initialize network parameters $\{\boldsymbol{\Theta}^{(l)}\}_{l=1}^{L}$ and view weight $\boldsymbol{\alpha}$;
while Do not converge or reach the maximum iteration number **do**
  Update network layers $\{\mathbf{H}^{(l)}\}_{l=1}^{L}$ using Eq. (5); // Forward propagation.
  Update network parameters $\{\boldsymbol{\Theta}^{(l)}\}_{l=1}^{L}$ and view weight $\boldsymbol{\alpha}$ using the gradients of the loss $\mathcal{L}_{cla}$; // Back propagation.
end while
Compute labels by $\widehat{\mathbf{y}}_i = \arg\max_j \mathbf{H}_{ij}^{(L)}$ for any i in $\{n+1, \cdots, N\}$;
Return the predictive label set $\{\widehat{\mathbf{y}}_i\}_{i=n+1}^{N}$.

4. Experiments

4.1. Dataset Descriptions

In this subsection, we give a brief description of eight real-world datasets containing different kinds of data types in Table 2. These datasets are described as follows:

- ALOI[1]: The dataset includes images captured under varied light conditions or rotation angles. Each image is associated with four feature sources, including 64-D RGB color histogram, 64-D HSV color histogram, 77-D color similarities features, and 13-D Haralick features.
- Animals[2]: This dataset contains 10,158 images categorized into 50 classes. Each image is accompanied by two extracted visual features: 4,096-D DECAF features and 4,096-D VGG19 features.
- Caltech102[3]: It is a dataset consisting of 9,144 pictures grouped into 102 categories, including 48-D Gabor features, 49-D WM features, 254-D GENTRIST features, 1,984-D HOG features, 512-D GIST features, and 928-D LBP features.
- GRAZ02[4]: This widely used object dataset comprises images from four different classes and includes six common features: 512-D GIST features, 225-D WT features, 256-D LBP features, 500-D SIFT features, 500-D SURF features, and 680-D PHOG features.
- Reuters[5]: This dataset consists of 18,758 documents in 6 categories, with multi-view information created from different languages, including English, French, German, Italian, and Spanish.
- OutScene[6]: This image dataset contains 2,688 instances categorized into eight classes. It includes 512-D GIST features, 59-D LBP features, 864-D HOG features, and 254-D GENT features.
- Scene15[7]: This scene image dataset comprises 4,485 images into 15 categories, with three views for each image. The feature dimensions for each perspective are 1,800, 1,180, and 1,240 respectively.
- Youtube[8]: This video dataset comprises 2,000 instances in 10 classes, with six views of both visual and audio features. The views include 2,000-D cuboids histogram, 1,024-D motion estimate histogram, 64-D HOG features, 512-D MFCC features, 64-D volume streams, and 647-D spectrogram streams.

[1] http://aloi.science.uva.nl
[2] http://attributes.kyb.tuebingen.mpg.de
[3] http://www.vision.caltech.edu/Image_Datasets/Caltech102/
[4] http://www.emt.tugraz.at/~pinz/data/GRAZ_02
[5] https://archive.ics.uci.edu/ml/datasets.html
[6] https://gitee.com/zhangfk/multi-view-dataset
[7] https://figshare.com/articles/15-Scene_Image_Dataset/7007177
[8] http://archive.ics.uci.edu/ml/datasets

Table 2: A brief description of the tested datasets.

Datasets	# Samples	# Features	# Views	# Classes	# Data types
GRAZ02	1,476	512 / 32 / 256 / 500 / 500 / 680	6	4	Object image
ALOI	1,079	64 / 64 / 77 / 13	4	10	Object image
Youtube	2,000	2,000 / 1,024 / 64 / 512 / 64 / 647	6	10	Object image
Animals	10,158	4,096 / 4,096	2	50	Animal image
Caltech102	9,144	48 / 40 / 254 / 1,984 / 512 / 98	6	102	Digit image
Scene15	4,485	1,800 / 1,180 / 1,240	3	15	Object image
OutScene	2,688	512 / 432 / 256 / 48	4	8	Object image
Reuters	18,758	21,531 / 24,892 / 34,251 / 15,506 / 11,574	5	6	Textual document

4.2. Compared Methods

We compare the performance of the proposed framework with the state-of-the-art models, including AMGL [41], MVAR [42], HLR-M^{2}VS [43], WREG [44], GCN-Fusion, Co-GCN [25], DSRL [45], LGCN-FF [26], and IMvGCN [28]. A description of these methods is given below.

- AMGL: It allows for the learning of optimal weights for each view without introducing any additional parameters.
- MVAR: It utilizes the $\ell_{2,1}$ norm to compute the regression loss for each independent view. It then constructs the objective function by taking the weighted sum of all the regression losses.
- HLR-M^{2}VS: It constructs a unified tensor space to jointly explore the relationships between multiple views through a local geometric structure. It uses low-level tensor regularization to ensure agreement across all views.
- WREG: It maps the concatenation of raw features to a discriminative low-dimensional subspace to integrate multi-view data. The features from different views are adaptively assigned to optimal weights, thus preserving both consistent and complementary information simultaneously.
- GCN-Fusion: It modifies the initial version of GCN which is unable to handle multi-view data directly. In order to address this limitation, it performs graph convolution by averaging the adjacency matrices and concatenating the feature matrices.
- Co-GCN: It introduces GCN into multi-view learning and obtains multi-view spectral information by adaptively combining Laplacian matrices.
- DSRL: It uses a deep sparse regularizer learning model to adaptively learn data-driven sparse regularizers for multi-view clustering and semi-supervised classification.

- LGCN-FF: It integrates sparse autoencoders and a learnable GCN to collectively learn comprehensive representations of multiple features and graphs.

- IMvGCN: It introduces multi-view reconstruction errors paired with Laplacian embeddings to capture independence and consistency.

4.3. Parameter Settings

Most of the compared algorithms in our study are implemented with their default parameters. In particular, certain parameter settings for the compared methods have been empirically determined to yield better performance, as follows:

- AMGL: It is a parameter-free framework, so we do not set any parameters for it.

- MVAR: We tune the trade-off weight for each view as $\lambda = 1000$, and fix the redistribution parameter r over views as 2.

- WREG: We select the trade-off λ in $\{1e^{-3}, 1e^{-2}, 1e^{-1}, 1e^{0}, 1e^{1}, 1e^{2}, 1e^{3}\}$ and the termination parameter $\epsilon = 0.001$.

- HLR-M^{2}VS: We select the weighted factors as $\lambda_1 = 0.2$ and $\lambda_2 = 0.4$. The maximum number of iterations is set to be 100.

- GCN-Fusion: We use the GCN-fusion method with a varying number of neighbors selected from $\{15, 20, 30, 50\}$.

- Co-GCN: The settings for the convolutional layers and number of neighbors are the same as those used in GCN-Fusion.

- DSRL: We set the block number as 10 and the initialization for the parameterized activation function is tuned as $w_1 = w_2 = 1$, $b_1 = 1$ and $b_2 = 2$.

- LGCN-FF: The default setting for the hyperparameter controlling the sparsity penalty degree is $\beta = 1$.

- IMvGCN: The default setting for hyperparameters $\lambda = 0.5$ and $\alpha = 1e^{-5}$.

For the GLGCN method, when handling data with feature dimensions exceeding 10,000, we specify the kernel matrix parameter of the diffusion map as $\sigma = 0.5$. The dimensions of the embedding space are determined using $\delta = 0.2$ and $k = 1$. Conversely, for datasets with lower feature dimensions, the kernel matrix parameter of the diffusion map is set to $\sigma = 1$. In this case, $\delta = 0.01$ and $k = 1$ are employed to ascertain the dimensionality of the embedding space.

A common threshold of $\tau = 0.001$ is used to sparsify the adjacency matrix for all datasets. To learn the graph representation, we employ a two-layer GCN with the ReLU activation function and optimize the model parameters using the Adam optimizer with a learning rate of 0.01 and a weight decay of $5e^{-8}$. The proposed framework is implemented using the PyTorch platform and runs on a computer featuring an AMD R9-5900X CPU, Nvidia RTX 3060 GPU, and 48GB of RAM.

4.4. Semi-Supervised Classification

In this subsection, we compare the performance of different methods by training each method with 10% of the available data and then testing them with the remaining 90%. To ensure the reliability of the results, we repeat each method five times and record their means and standard deviations. We utilize accuracy and F1-score as the criteria for evaluation, as shown in Table 3. Based on the experimental results, we have the following observations.

Table 3: Classifiation results (mean% and standard deviation%) of all compared semi-supervised classification methods with 10% labeled samples as supervision, where the best results are highlighted in bold and the second best results are highlighted in underlined.

Dataset	Metrics	AMGL	MVAR	WREG	HLR-M^2VS	GCN-Fusion	Co-GCN	DSRL	LGCN-FF	IMvGCN	GLGCN
ALOI	ACC	56.81 (1.43)	30.78 (10.67)	90.19 (1.33)	89.73 (0.78)	90.27 (1.74)	79.97 (1.98)	90.91 (0.85)	95.67 (0.65)	77.70 (3.80)	**97.25 (0.42)**
	F1	57.71 (3.02)	23.94 (10.67)	90.70 (1.21)	90.14 (0.75)	90.71 (0.74)	79.13 (2.44)	91.00 (0.79)	95.64 (0.13)	76.10 (5.10)	**97.37 (0.01)**
Animals	ACC	70.96 (0.49)	81.51 (0.47)	**83.51 (0.34)**	72.73 (0.53)	77.38 (0.57)	80.20 (1.22)	80.00 (0.50)	54.98 (4.78)	81.91 (0.08)	83.25 (0.19)
	F1	65.69 (0.66)	76.69 (1.04)	**78.65 (0.49)**	68.05 (0.90)	74.03 (0.72)	73.74 (1.56)	73.62 (1.14)	40.20 (6.13)	75.75 (0.17)	78.37 (0.31)
Caltech102	ACC	46.72 (0.47)	46.13 (0.90)	46.93 (0.55)	48.07 (0.44)	46.14 (0.62)	37.98 (8.71)	52.88 (0.56)	40.16 (0.79)	47.60 (0.10)	**53.59 (0.09)**
	F1	30.33 (0.66)	29.02 (0.95)	28.53 (0.82)	31.18 (0.74)	27.13 (0.24)	20.91 (6.40)	**34.57(1.24)**	33.42 (0.46)	24.30 (0.10)	34.21 (0.27)
GRAZ02	ACC	54.95 (1.00)	52.46 (1.77)	43.40 (3.52)	54.69 (2.61)	55.67 (6.20)	40.54 (2.56)	48.11 (1.04)	49.62 (2.50)	56.19 (0.49)	**61.97 (0.52)**
	F1	55.60 (1.05)	52.94 (1.98)	43.55 (3.58)	56.32 (1.78)	58.68 (0.91)	38.94 (1.50)	48.64 (1.05)	43.67 (0.96)	55.00 (0.55)	**61.02 (0.39)**
Scene15	ACC	68.41 (0.66)	44.25 (9.65)	52.32 (1.95)	67.40 (1.34)	72.69 (0.66)	58.67 (1.09)	61.75 (0.85)	50.05 (4.38)	65.56 (3.05)	**73.47 (0.26)**
	F1	67.30 (0.70)	45.83 (8.41)	52.58 (1.99)	67.3 (0.86)	72.40 (0.41)	56.69 (0.89)	60.54 (0.82)	42.32 (5.71)	62.03 (2.93)	**72.41 (0.29)**
Youtube	ACC	49.27 (1.00)	22.51 (2.71)	36.72 (0.39)	55.50 (0.00)	55.89 (1.34)	29.28 (0.27)	44.74 (0.80)	47.30 (1.84)	47.20 (0.60)	**59.30 (0.48)**
	F1	48.83 (1.02)	20.39 (2.04)	35.91 (0.92)	51.64 (2.05)	55.92 (1.20)	21.53 (1.28)	42.11 (2.94)	42.32 (5.71)	45.70 (0.60)	**59.00 (0.63)**
OutScene	ACC	71.16 (0.98)	46.10 (11.26)	57.63 (1.80)	73.33 (1.25)	75.23 (0.73)	70.96 (2.05)	44.74 (0.80)	61.06 (11.03)	77.20 (0.72)	**77.36 (0.29)**
	F1	72.30 (0.76)	50.81 (11.09)	58.61 (1.59)	75.23 (1.21)	75.62 (0.68)	71.31 (2.02)	42.11 (2.94)	57.94 (15.85)	**77.44 (0.78)**	77.37 (0.31)
Reuters	ACC	OM	64.60 (0.30)	OM	OM	OM	60.20 (0.60)	OM	67.30 (0.50)	65.96 (1.98)	**71.50 (0.99)**
	F1	OM	60.72 (0.34)	OM	OM	OM	56.39 (1.23)	OM	64.73 (1.18)	61.03 (1.70)	**65.82 (1.46)**

First of all, the proposed model exhibits satisfactory performance on most datasets, outperforming other baseline methods. Significantly, our model demonstrates notable improvements on the GRAZ02, Youtube, and Reuters datasets, surpassing the second-highest performing algorithm by 6.3%, 3.4%, and 4.2%, respectively. These observations validate the effectiveness of the proposed framework.

Figure 2 displays the changes in training loss and test accuracy as the model trains on the six datasets. From the figure, it is evident that the loss and accuracy of all datasets converge after 200 epochs, with the accuracy gradually increasing as the loss decreases. This also demonstrates the effectiveness of the proposed model, which requires only a few rounds of training to achieve both high classification accuracy and low loss.

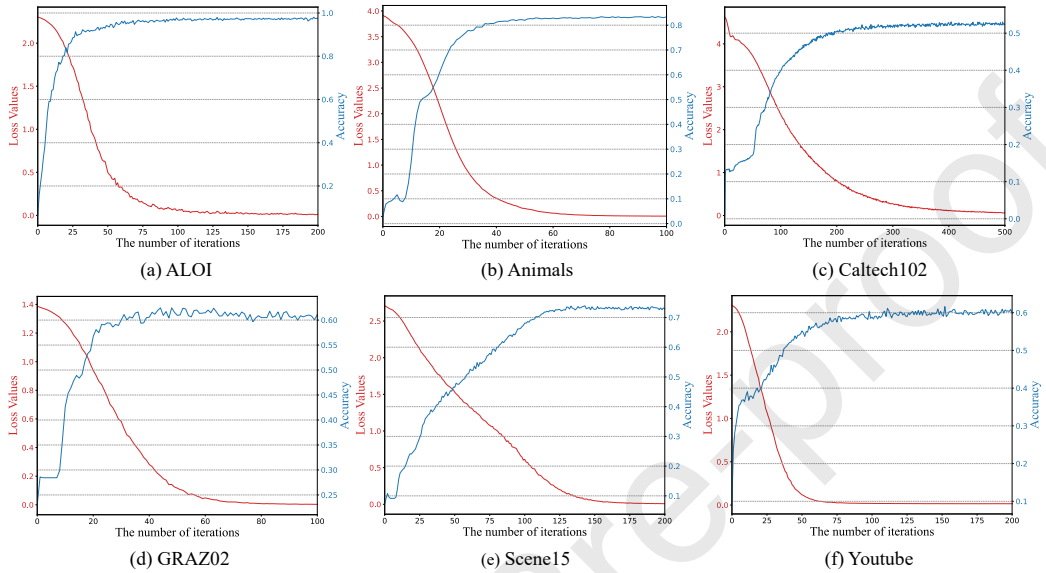


Figure 2: Convergence curves of training loss values and test accuracy with GLGCN on six datasets.

Table 4 shows the dimensions of the original data and the reduced dimensions after applying the diffusion map. These results confirm that we obtain lower dimensional features with a global structure by reducing the dimensionality of the data at a given scale. To summarize, the proposed model can learn a low-dimensional representation of the original data and achieve satisfactory performance with reliable convergence.

Additionally, the performance of each compared algorithm at different labeled sample ratios is illustrated in Figure 3. Based on the figure, it shows that GLGCN is better suitable for semi-supervised classification tasks as it achieves satisfactory performance even with a low sample label rate. In comparison, other methods require significantly more supervised information to match its level of performance.

To better showcase the performance of the proposed model, Figure 4 presents the visualization results obtained by each algorithm using t-SNE on the ALOI dataset. The figure demonstrates that GLGCN assigns more accurate class labels, and it has more distinct inter-class demarcation and better inter-class separability.

Table 4: Comparison of original feature dimensions and feature dimensions after diffusion map.

Dataset	Original Dimensions	Dimensions after Diffusion Map
ALOI	64 / 64 / 77 / 13	64 / 56 / 4 / 7
Animals	4,096 / 4,096	1,451 / 649
Caltech102	48 / 40 / 254 / 1,984 / 512 / 928	29 / 26 / 121 / 226 / 126 / 116
GRAZ02	512 / 32 / 256 / 500 / 680	61 / 32 / 21 / 126 / 500 / 212
Scene15	1,800 / 1,180 / 1,240	581 / 42 / 492
Youtube	2,000 / 1,024 / 64 / 512 / 64 / 647	414 / 247 / 45 / 311 / 63 / 346
OutScene	512 / 432 / 256 / 48	70 / 38 / 37 / 48
Reuters	21,531 / 24,892 / 34,251 / 15,506 / 11,574	15 / 12 / 14 / 12 / 11

In a nutshell, our proposed model yields more accurate results with less supervision and is able to more effectively distinguish between different classes of sample clusters.

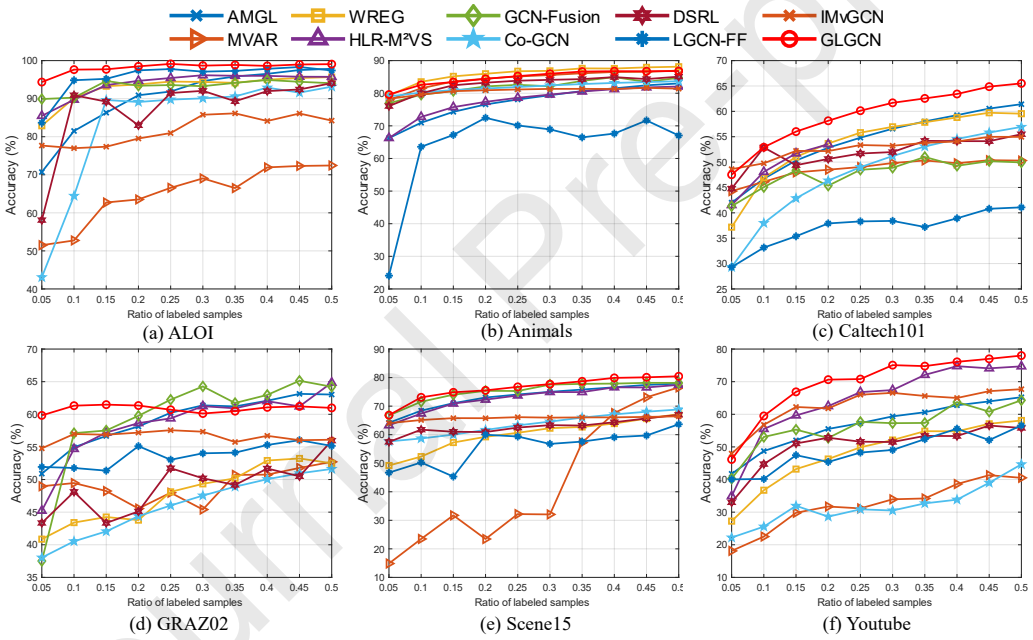


Figure 3: The various performance (Accuracy %) of all compared methods on six test datasets. The ratio of labeled samples ranges in $\{0.05, 0.10, ..., 0.50\}$.

4.5. Feature Missing Analysis

To demonstrate the stability of our model, we apply a random feature mask to the multi-view data. This may lead to the possibility of connecting samples

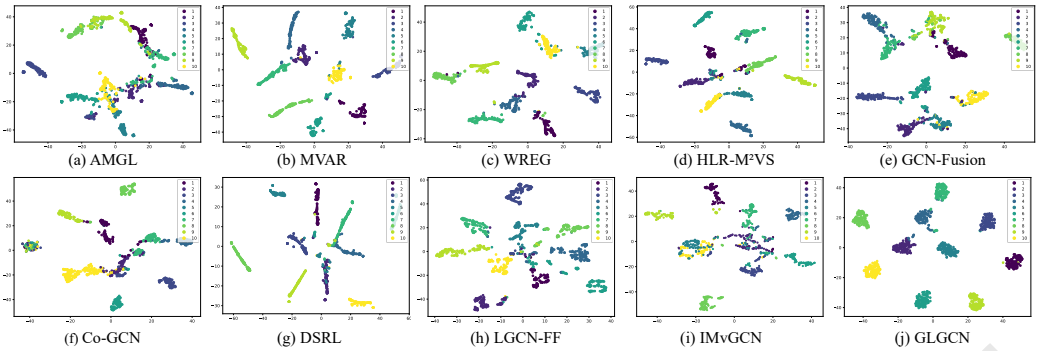


Figure 4: T-SNE visualization of semi-supervised classification results from all compared methods on the dataset ALOI.

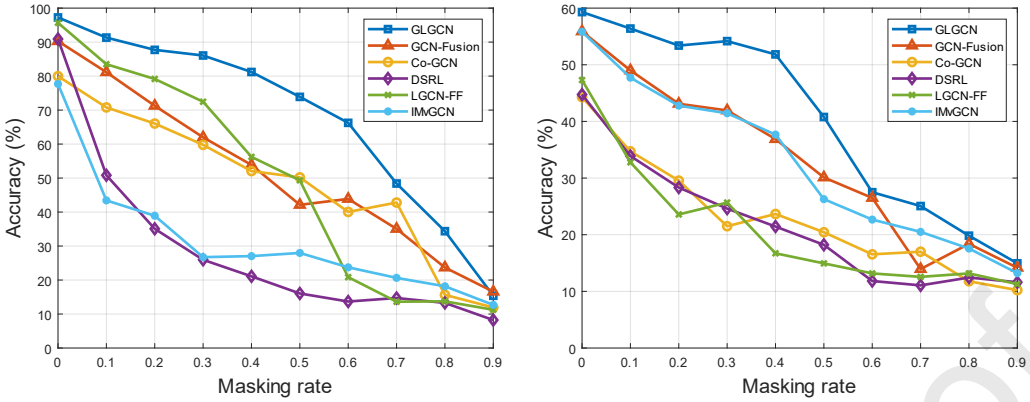
that are not originally close to each other. Given that the features of the datasets are initially complete, we apply a masking process to obtain an incomplete feature matrix as follows: randomly generate a matrix of the same size as the feature matrix, containing elements of 0 or 1. The final incomplete matrix is obtained by performing a Hadamard product between the masked matrix and the feature matrix, where the masking rate is determined by the proportion of zeros in the masked matrix to the total number of elements in the entire matrix. Figure 5 shows the performance of the GCN-based algorithm as the size of the masking rate increases. It can be observed that the performance of all algorithms decreases as the feature masking rate increases, but GLGCN shows slower performance degradation compared to other algorithms. Additionally, the proposed framework is able to withstand a larger masking rate while achieving comparable performance to other algorithms. This further demonstrates the effectiveness of the topological connection constructed using diffusion distance.

4.6. Ablation Study

We also test the effectiveness of constructing adjacency matrices by diffusion map and the learnable weights. The results of the ablation experiments are shown in Table 5. We can observe that the best results are achieved by using both the diffusion map and the learnable weights. Moreover, using only one of these techniques yields better results than not using either of them. Based on the observation, we can conclude that both two components of GLGCN are beneficial for improving the performance of semi-supervised classification tasks.

4.7. Parameter Sensitivity Analysis

Figure 6 demonstrates how the model performance is affected by varying the values of σ and τ across different datasets. This is achieved by adjusting the



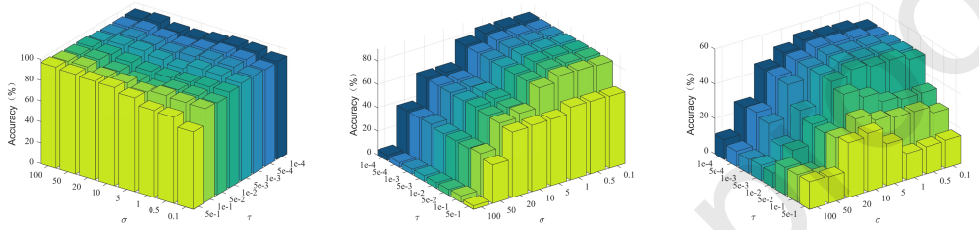
(a) ALOI (b) Youtube

Figure 5: The various performance (Accuracy %) of deep network-based methods on ALOI and Youtube. The masking rate ranges in $\{0.0, 0.1, \cdots, 0.9\}$.

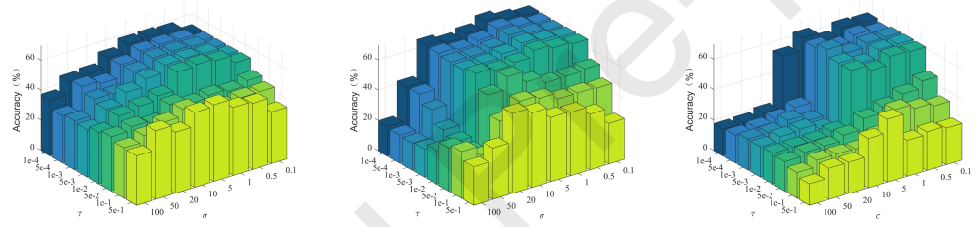
Table 5: Ablation study of the proposed GLGCN on all test datasets.

Dataset	$\psi_k^{(v)}(\cdot)$	α	ACC	Dataset	$\psi_k^{(v)}(\cdot)$	α	ACC
ALOI			90.11	GRAZ02			55.98
		✓	93.79			✓	60.72
	✓		94.42		✓		60.95
	✓	✓	**97.25**		✓	✓	**61.97**
Animals			76.15	Scene15			69.40
		✓	78.82			✓	72.62
	✓		82.03		✓		72.99
	✓	✓	**83.25**		✓	✓	**73.47**
Caltech102			47.73	Youtube			53.11
		✓	48.09			✓	56.66
	✓		52.51		✓		57.57
	✓	✓	**53.59**		✓	✓	**59.30**
Reuters			-	OutScene			75.23
		✓	-			✓	75.99
	✓		65.18		✓		75.94
	✓	✓	**71.50**		✓	✓	**77.36**

parameters of the Gaussian kernel function, which alters the similarity of the original samples after being mapped by the kernel function, and the parameters of the truncated diffusion correlation function, which modifies the distance threshold required to establish connections between the samples. The experimental results reveal that, for these datasets, the proposed model tends to perform better with smaller values of σ and τ. Remarkably, there is not much variation in the performance of the model for the dataset ALOI when these two parameters are modified. This could be attributed to the inherent perfect geometric structure of the dataset, which remains relatively unchanged after diffusion map at various scales, and the stability of sample distances.



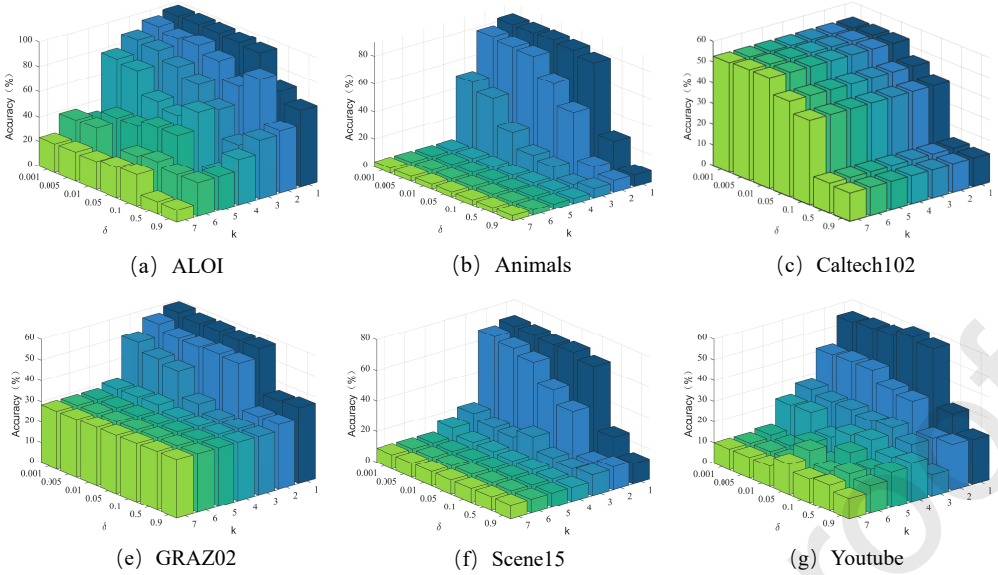
(a) ALOI (b) Animals (c) Caltech102



(d) GRAZ02 (e) Scene15 (f) Youtube

Figure 6: Parameter sensitivity (Accuracy%) of the proposed method w.r.t. σ and τ on six datasets.

It can be observed from Figure 7 that the values of threshold parameter δ and time step k have a significant impact on the accuracy of the model. The optimal performance is achieved when setting $k = 1$ and $\delta = 0.01$. This suggests that the proposed method is sensitive to the choice of these two parameters, and parameter fine-tuning is necessary to achieve the best results. Generally, the performance gradually decreases as k increases, while it improves with decreasing δ at a fixed k value. Notably, when k exceeds 3, increasing δ has little effect on the performance. It is also interesting to observe that the performance of the Caltech102 dataset is primarily affected by changes in δ, while variations in the k value have minimal effect. One possible explanation for this is that when keeping δ fixed, the neighborhoods constructed based on diffusion distances do not change



(a) ALOI (b) Animals (c) Caltech102

(e) GRAZ02 (f) Scene15 (g) Youtube

Figure 7: Parameter sensitivity (Accuracy%) of the proposed method w.r.t. γ and k on six datasets.

significantly regardless of how k is adjusted, indicating that the selected scale σ sufficiently captures the underlying geometric structure of the dataset to create stable adjacency matrices.

5. Conclusion

This paper proposed a framework for learning robust multi-view distance metrics by considering the limitations of similarity measures constructed in the original feature space, which may ignore noise in the features and the underlying geometric structure. To accomplish this goal, we leveraged diffusion maps to capture a more precise global structure in feature space and preserved the nonlinear distance relationships between the data points. In constructing the distance relationships between samples, the proposed method did not rely on node-level distances to determine neighborhoods. Instead, we employed a perspective-level approach to determine which distances were valid for the entire perspective and encoded these relationships as adjacency matrix weights. To perform feature fusion, we applied learnable weights to the features of each perspective prior to concatenating them. The experimental results on benchmark datasets clearly demonstrated the superior performance of our proposed framework compared to other state-of-the-art methods for semi-supervised classification tasks.

There are several promising topics in the realm of multi-view learning and GCNs that have yet to be explored. Many existing studies operate under the assumption of reliable original data and overlook the significance of local relationships among samples, particularly in the context of GCN-based models. In the future, our objective is to advance the topological construction process of GLGCN by incorporating downstream tasks, thereby constructing graphs that are better tailored to the specific requirements of those tasks.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grant U21A20472 and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503.

References

[1] G. Li, D. Song, W. Bai, K. Han, R. Tharmarasa, Consensus and complementary regularized non-negative matrix factorization for multi-view image clustering, Information Sciences 623 (2023) 524–538.

[2] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, P. Fua, Learning monocular 3d human pose estimation from multi-view images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8437–8446.

[3] Y. Zhang, J. Wu, Z. Cai, S. Y. Philip, Multi-view multi-label learning with sparse feature selection for image annotation, IEEE Transactions on Multimedia 22 (2020) 2844–2857.

[4] M. Liu, Y. Wang, V. Palade, Z. Ji, Multi-view subspace clustering network with block diagonal and diverse representation, Information Sciences 626 (2023) 149–165.

[5] J. Cheng, Q. Wang, Z. Tao, D. Xie, Q. Gao, Multi-view attribute graph convolution networks for clustering, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 2973–2979.

[6] Z. Fang, S. Du, X. Lin, J. Yang, S. Wang, Y. Shi, Dbo-net: Differentiable bi-level optimization network for multi-view clustering, Information Sciences 626 (2023) 572–585.

[7] Y. Tian, S. Sun, J. Tang, Multi-view teacher–student network, Neural Networks 146 (2022) 69–84.

[8] X. Xu, W. Li, D. Xu, I. W. Tsang, Co-labeling for multi-view weakly labeled learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2015) 1113–1125.

[9] W. Guo, Z. Wang, W. Du, Robust semi-supervised multi-view graph learning with sharable and individual structure, Pattern Recognition 140 (2023) 109565.

[10] S. N. Satchidanand, H. Ananthapadmanaban, B. Ravindran, Extended discriminative random walk: A hypergraph approach to multi-view multi-relational transductive learning, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 3791–3797.

[11] Y. Wang, X. Lin, Q. Zhang, Towards metric fusion on multi-view data: a cross-view based graph random walk approach, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 805–810.

[12] K. Luong, R. Nayak, T. Balasubramaniam, M. A. Bashar, Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering, Pattern Recognition 131 (2022) 108815.

[13] M. Zhao, W. Yang, F. Nie, Auto-weighted orthogonal and nonnegative graph reconstruction for multi-view clustering, Information Sciences 632 (2023) 324–339.

[14] M. R. Khan, J. E. Blumenstock, Multi-gcn: Graph convolutional networks for multi-view networks, with applications to global poverty, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 606–613.

[15] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, 2017, pp. 1–13.

[16] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: Proceedings of the 7th International Conference on Learning Representations, 2019, pp. 1–13.

[17] F. M. Bianchi, D. Grattarola, C. Alippi, Spectral clustering with graph neural networks for graph pooling, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 874–883.

[18] J. Lee, I. Lee, J. Kang, Self-attention graph pooling, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 3734–3743.

[19] J. Chen, H. He, F. Wu, J. Wang, Topology-aware correlations between relations for inductive link prediction in knowledge graphs, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 6271–6278.

[20] M. Zhang, Y. Chen, Link prediction based on graph neural networks, in: Proceedings of the 32nd Conference on Neural Information Processing Systems, 2018, pp. 5171–5181.

[21] Z. Zhu, Z. Zhang, L.-P. Xhonneux, J. Tang, Neural bellman-ford networks: A general graph neural network framework for link prediction, in: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021, pp. 29476–29490.

[22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.

[23] J. Chang, C. Gao, X. He, D. Jin, Y. Li, Bundle recommendation with graph convolutional networks, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1673–1676.

[24] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.

[25] S. Li, W. Li, W. Wang, Co-gcn for multi-view semi-supervised learning, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 4691–4698.

[26] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, S. Wang, Learnable graph convolutional network and feature fusion for multi-view learning, Information Fusion 95 (2023) 109–119.

[27] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, J. Pei, AM-GCN: adaptive multi-channel graph convolutional networks, in: Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2020, pp. 1243–1253.

[28] Z. Wu, X. Lin, Z. Lin, Z. Chen, Y. Bai, S. Wang, Interpretable graph convolutional network for multi-view semi-supervised learning, IEEE Transactions on Multimedia (2023) 1–14.

[29] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017, pp. 2921–2927.

[30] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, J. Yin, Subspace segmentation-based robust multiple kernel clustering, Information Fusion 53 (2020) 145–154.

[31] X. Jia, X.-Y. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, D. Yue, Semi-supervised multi-view deep discriminant representation learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2020) 2496–2509.

[32] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: Proceedings of the 2nd International Conference on Learning Representations, 2014, pp. 1–12.

[33] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of the 30th Conference on Neural Information Processing Systems, 2016, pp. 3837–3845.

[34] J. Gasteiger, S. Weißenberger, S. Günnemann, Diffusion improves graph learning, in: Proceedings of the 33rd Conference on Neural Information Processing Systems, 2019, pp. 13333–13345.

[35] Y. You, T. Chen, Z. Wang, Y. Shen, L2-GCN: layer-wise and learned efficient training of graph convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2124–2132.

[36] G. Yang, J. Cao, Q. Sheng, P. Qi, X. Li, J. Li, DRAG: dynamic region-aware GCN for privacy-leaking image detection, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022, pp. 12217–12225.

[37] F. Wu, X.-Y. Jing, P. Wei, C. Lan, Y. Ji, G.-P. Jiang, Q. Huang, Semi-supervised multi-view graph convolutional networks with application to webpage classification, Information Sciences 591 (2022) 142–154.

[38] B. Jiang, C. Zhang, Y. Zhong, Y. Liu, Y. Zhang, X. Wu, W. Sheng, Adaptive collaborative fusion for multi-view semi-supervised classification, Information Fusion 96 (2023) 37–50.

[39] C. Zhang, H. Fu, Q. Hu, P. Zhu, X. Cao, Flexible multi-view dimensionality co-reduction, IEEE Transactions on Image Processing 26 (2017) 648–659.

[40] J. Yuan, K. Gao, P. Zhu, K. O. Egiazarian, Multi-view predictive latent space learning, Pattern Recognition Letters 132 (2020) 56–61.

[41] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 1881–1887.

[42] H. Tao, C. Hou, F. Nie, J. Zhu, D. Yi, Scalable multi-view semi-supervised classification via adaptive regression, IEEE Transactions on Image Processing 26 (2017) 4283–4296.

[43] Y. Xie, W. Zhang, Y. Qu, L. Dai, D. Tao, Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning, IEEE Transactions on Cybernetics 50 (2020) 572–586.

[44] M. Yang, C. Deng, F. Nie, Adaptive-weighting discriminative regression for multi-view classification, Pattern Recognition 88 (2019) 236–245.

[45] S. Wang, Z. Chen, S. Du, Z. Lin, Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 5042–5055.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Shiping Wang reports financial support was provided by National Natural Science Foundation of China.