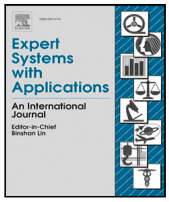
Expert Systems With Applications 238 (2024) 122151

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Review

GAF-Net: Graph attention fusion network for multi-view semi-supervised classification

Na Song [a,b], Shide Du [c,d], Zhihao Wu [c,d], Luying Zhong [c,d], Laurence T. Yang [a,e], Jing Yang [a], Shiping Wang [c,d,*]

[a] School of Computer Science and Technology, Hainan University, Haikou 570228, China
[b] School of Mechanical, Electrical, and Information Engineering, Putian University, Putian 351100, China
[c] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[d] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350108, China
[e] Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada

ARTICLE INFO

Keywords:
Multi-view learning
Multi-modal fusion
Semi-supervised classification
Graph attention
Graph neural network

ABSTRACT

Multi-view semi-supervised classification is a typical task to classify data using a small amount of supervised information, which has attracted a lot of attention from researchers in recent years. In practice, existing methods tend to focus on extracting spatial or spectral features using graph neural networks without considering the diversity and variability of graph structures and the contributions of different views. To address this challenge, a framework termed graph attention fusion network is proposed, which consists of two phases: view-specific feature embedding and graph embedding fusion. In the former feature extraction stage, the view-specific feature embedding module can flexibly focus on the neighborhood calculation operation to learn a weight for each neighboring node. In the latter feature fusion stage, the graph embedding fusion module is performed by complementarity and consistency to fuse these embeddings for semi-supervised classification tasks. We carry out comprehensive experiments in semi-supervised classification on real-world datasets to substantiate the effectiveness of the proposed approach compared to several existing state-of-the-art methods.

1. Introduction

In recent decades, multi-view representation learning has been extensively investigated as a significant multimedia technology. In this context, multimedia technology has been broadly applied to natural language understanding (Hirschberg & Manning, 2015; Otter et al., 2021), social networks (Musetti et al., 2022; Nie, Song et al., 2022), computer vision (Chai et al., 2021; Nie, Qu et al., 2022), recommendation systems (Chen & Wang, 2022; Chen, Zhao et al., 2021), medical diagnosis (Han, Yang et al., 2022) and other fields. Specifically, multi-view data can be represented in many forms as a result of the same object captured by different sensor devices. For instance, an entity can be described through an image, an audio, or a video (Deng et al., 2019; Zhang et al., 2022). Meanwhile, most of the multi-view data are highly redundant (Gan & Ma, 2022; Xu et al., 2020; Zhao et al., 2023), and how to process such data becomes an urgent task (Chen, Huang et al., 2021; Han, Ren et al., 2022; Wang, Chen et al., 2022). As a consequence, multi-view learning has emerged, and multi-view semi-supervised classification has been widely used in various fields as one of the essential applications (Chen, Cao et al., 2022; Pan & Kang, 2021; Wang, Wang et al., 2021; Xia et al., 2022).

Multi-view semi-supervised classification exploits the consistency and complementarity of multi-view features obtained from heterogeneous sources to learn common features, and has gained widespread attention and applications (Chen, Liu et al., 2022; Fu et al., 2022; Zhang et al., 2018). For the past few years, a large amount of excellent multi-view semi-supervised classification and clustering methods have emerged. These approaches consider the correlation between multiple views and less complementary information in feature learning (Nie et al., 2016), subspace learning (Yang et al., 2019), attributed graph learning (Lin et al., 2023) and collaborative training (Xie et al., 2020). GCN-based methods have made great progress in node classification for multi-view data. However, most of these models are based on fixed pre-constructed adjacency matrices. In the model proposed by (Yao et al. 2022), the current graph topology is assumed to be unknown. By using an attention-based feature fusion mechanism to fuse complementary information from multiple views, a better graph representation can be

* Corresponding author at: College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China.
E-mail addresses: nasong1010@163.com (N. Song), dushidems@gmail.com (S. Du), zhihaowu1999@gmail.com (Z. Wu), luyingzhongfzu@163.com (L. Zhong), ltyang@gmail.com (L.T. Yang), jingyang@hainanu.edu.cn (J. Yang), shipingwangphd@163.com (S. Wang).

https://doi.org/10.1016/j.eswa.2023.122151
Received 29 March 2023; Received in revised form 10 October 2023; Accepted 11 October 2023
Available online 19 October 2023

obtained. Furthermore, the utilization of reasonable feature representation methods generally improves the performance when fusing multiple views, but some problems such as overbalancing of each view and overdependence on parameters could occur when these methods are combined with weight learning (Xia et al., 2021). These under-explored problems such as relatively heavy training time and the tendency to fall into local optimality are equally important (Fei-Fei et al., 2004; Tang et al., 2022). Therefore, many methods based on GCN variants have received attention and development, and achieved encouraging progress. Moreover, the GCN-based approach can investigate consistent information among multiple perspectives in greater depth and offer supplementary details for views with significant disparities. In particular, GCN is intended for analyzing graph-structured data in the context of spectral theory. It boosts impressive graph representation abilities, which have been shown to be frequently superior to those of alternative approaches in empirical studies.

Although these methods based on GCN have achieved encouraging performance, they suffer from an over-reliance on graph structures and an unbalanced weight of edges. GCN-based methods over-rely on the graph structure because they operate directly on the adjacency matrix that represents the graph structure. In the case of a single view, without considering the relationships between views, the GCN-based approach may produce incorrect results when the graph structure is incomplete or noisy. To mitigate this, we propose a network that introduces multiple view features to help GCN capture more comprehensive embeddings beyond the graph structure. We incorporate node attributes from a single view and relationships between multiple views into the graph convolution operation using attention mechanisms, enabling the model to consider both node information and structural features simultaneously. Moreover, through self-attention mechanisms, we dynamically capture dependencies within and across views, reducing over-dependence on specific graph structures and improving model generalization and robustness. It is a common issue, especially in multi-view data where each view contributes to the final fused representation, but its contribution varies and there is a certain amount of involved noise. Therefore, our aim is to assign larger weights to those views that contribute more to the consistency representation, thus constructing a trustworthy fusion representation for the semi-supervised classification task. To tackle these challenges, the objective of this paper is to construct a network architecture that adequately adapts to various graph structural features for network training and to develop an efficient weight calculation method. Therefore, we propose a framework that not only handles complex graph structure and lessens the dependence on the graph structure, but also applies it to diverse graph nodes by assigning appropriate weights to the connected nodes. The overall framework diagram is illustrated in Fig. 1. In the feature extraction stage, the view-specific feature embedding module can focus more flexibly on the calculation operation of the neighborhood to learn the weight coefficients of each neighbor. During the feature fusion stage, the graph embedding fusion module complements and maintains consistency in fusing these embeddings for semi-supervised classification tasks.

The main contributions of this paper can be summarized in the following three aspects:

- Propose a graph attention fusion network to integrate a consistent embedding for a single view and an adaptive weight fusion for multiple views.
- Construct a two-stage projection that includes view-specific feature embedding and graph embedding fusion, where the former aims at mining the node-consistent features of heterogeneous graphs, while the latter focuses on fusing the complementary representation of multi-view data.
- Experimental results validate the inspiring performance of the proposed method with limited labeled data in terms of multi-view semi-supervised classification.

2. Related work

In this section, we review recent work on multi-view semi-supervised classification, graph convolutional networks, and attention mechanisms.

2.1. Multi-view semi-supervised classification

Semi-supervised classification is a major application direction of multi-view learning, especially in scenarios where data labeling is costly or difficult to obtain (Huang et al., 2022). In reality, semi-supervised learning methods are also utilized in those scenarios where there is no significant lack of labeled data, while unlabeled data are effortlessly obtained. It can potentially improve classification performance (Van Engelen & Hoos, 2020). For example, Wang, Wang and Guo (2021) proposed an accelerated embedding method that could solve the multi-view semi-supervised classification problem by automatically learning the optimal weight of each view through a small amount of labeled data. Liu et al. (2022) proposed an iterative framework with a support vector machine to complement uncertain view data by learning the consistency of multiple views. Wang, Fu et al. (2022) proposed a view-specific representation and class probability estimation method to concatenate multiple views by improving pseudo labels, and to further learning consistent classification information. Zhang et al. (2021) proposed a robust multi-view fusion model to enhance the label propagation capability and achieve expectation maximization, which is not limited by the angular interval of multiple views. Wang, Shen et al. (2022) proposed a supervised classification method incorporating weighted elasticity loss for the fusion of submodels using complementarity from all views and private information from a single view.

Numerous studies have recognized that the learning performance of multi-view algorithms is generally more delightful than that of single-view ones for semi-supervised classification tasks. However, the supervision rate of the multi-view semi-supervised learning algorithm has an impact on the classification accuracy in practical applications. Therefore, our method focuses on the features that contribute most to consistent embedding in the process of multi-view feature extraction and fusion to reduce the impact of the supervision rate on semi-supervised classification accuracy.

2.2. Graph convolutional networks

The graph convolutional network was first proposed by Kipf and Welling (2017) and experimentally validated to be more advantageous in citation networks and knowledge graph data application scenarios. A learnable semi-supervised feature representation learning method is proposed through a scalable graph convolution structure. We denote

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}, \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ represents the adjacency matrix of the graph, including self-loop. Here, $\mathbf{I}_N$ is identity matrix, and $\tilde{\mathbf{D}} = [\tilde{\mathbf{D}}_{ij}]_{N\times N}$ is a diagonal matrix with $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. In many cases, a classical two-layer GCN is used for the semi-supervised node classification task. Consequently, the forward propagation of a two-layer GCN model is expressed as

$$\mathbf{Z} = softmax(\hat{\mathbf{A}}ReLU(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}), \tag{2}$$

where $\mathbf{Z}$ is the output as the final feature representation. There is a large amount of work emerging from the fundamental GCN. For example, Wu et al. (2019) reduced the complexity of GCN with a fixed low-pass filter and a linear classifier. Li et al. (2020) extended the framework using Laplace operators and proposed a classification method for adaptively aggregating graph information of multi-view data. Chen et al. (2020) proposed a linear model for large-scale data

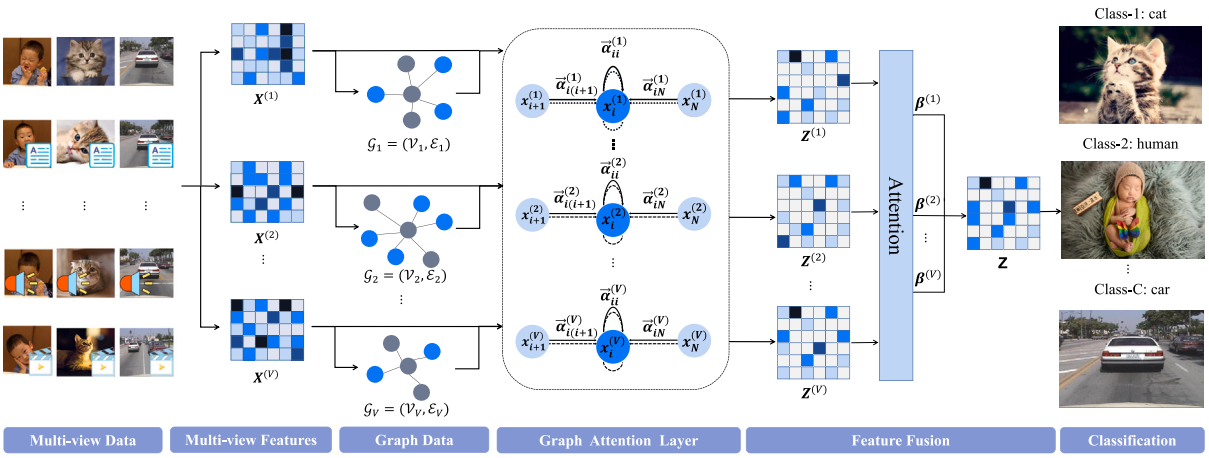


Fig. 1. An illustration of the proposed method, which performs a multi-view semi-supervised classification task by learning the embedding of a fused graph through a view-specific feature embedding module and a graph embedding fusion module.

to alleviate the over-smoothing problem during graph convolution aggregation using a residual network structure. Wang et al. (2020) introduced an adaptive weight learning framework to extract the most relevant embedding representation from node features for multi-channel network semi-supervised classification applications.

These prior works inspire us to employ a tailored GCN for multi-view semi-supervised classification tasks. The construction of the hierarchical attention mechanism of the GCN in the semi-supervised classification of undirected graphs is able to encode graph structures and node features appropriately.

2.3. Attention mechanism

The transformer model of the attention mechanism was proposed by Vaswani et al. (2017). The model consists of a set of queries, keys, and values, denoted by **Q**, **K**, and **V**. The output attention matrix is expressed as

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\mathbf{V}^T tanh(\mathbf{WQ}+\mathbf{UK})), \tag{3}$$

where **W** and **U** are learnable network parameters. The above formula is concatenated and projected to facilitate parallel execution of the attention mechanism, resulting in the multi-head attention as

$$\begin{aligned} MultiHead(\mathbf{Q},\mathbf{K},\mathbf{V}) &= Concat(\mathbf{head}_1,\ldots,\mathbf{head}_h)\mathbf{W}^O, \\ \text{where}\quad \mathbf{head}_i &= Attention(\mathbf{QW}_i^Q,\mathbf{KW}_i^K,\mathbf{VW}_i^V), \end{aligned} \tag{4}$$

where $\mathbf{W}^O$ denotes the multi-head learnable weight matrix throughout a linear layer, and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ are learnable weights for **Q**, **K**, and **V**.

Compared to the method of weights calculated by convolution, the attention-based method has also attracted increasing attention. For example, Veličković et al. (2017) proposed a graph neural network framework on a self-attentive mechanism, which assigned different weight coefficients to neighboring nodes that disregards the graph structure. Zt et al. (2020) constructed a multimodal attention network based on graph learning for personalized recommendation. Chen, Fragonara et al. (2021) extended the graph attention network to the use of both the node features in the graph and the edge features, and iterated them in a parallel manner. Guo et al. (2021) presented a target-tracking architecture applied to the graph attention mechanism to propagate the template data of the target information with the searched features.

Inspired by the ability of the attention-based mechanism to dynamically and adaptively discover relationships between nodes and reduce the complexity of the model, we use an improved attention-based graph convolutional network for the multi-view semi-supervised classification task.

3. The proposed method

We give a brief description of the whole process of the method. First, we describe the different view features of the same object using multiple graphs, where the nodes represent the features of a particular view and the edges represent the relationship between two features. Second, we focus on the discriminative feature representation of each node and its neighbors and extract the view-specific intrinsic information of each view in preparation for subsequent fusion. Finally, the extracted features of samples from different views are fused by the degree of correlation between them, discarding the features with a small contribution to the nodes and assigning larger weights to the features of the nodes with a greater contribution to the fusion.

To illustrate the use of mathematical symbols in this paper, Table 1 lists the explanations of elementary symbols. Note that superscripts indicate different views and subscripts represent different nodes.

3.1. Network module

In reality, there is both diversity and consistency among the multi-view descriptions of the same object. In general, node classification tasks require the extraction of feature representations from multiple views based on the consistency and complementarity between the multi-view features for downstream fusion and classification discrimination. Consequently, we need to search not only for the intrinsic features in the graph structure domain, but also for the correlated features between different heterogeneous views. To this end, we propose a multi-view graph attention network.

Formally, we treat the multi-view data as the input to the network, so the input as a node feature set is represented as $\{\mathbf{X}^{(v)} \in \mathbb{R}^{N\times D_{(v)}}\}_{v=1}^{V}$, where $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)};\mathbf{x}_2^{(v)};\cdots;\mathbf{x}_N^{(v)}]$. Here, V is the number of views, $D_{(v)}$ is the feature dimension of the vth view data, and N is the number of samples. $\{\mathbf{x}_1^{(v)};\cdots;\mathbf{x}_n^{(v)}\}_{v=1}^{V}$ denotes labeled samples and the rest are unlabeled samples. The output of the layer is represented as $\{\mathbf{Z}^{(v)} \in \mathbb{R}^{N\times D'_{(v)}}\}_{v=1}^{V}$, where $D'_{(v)}$ denotes the output feature dimension.

3.1.1. View-specific feature embedding module

Considering that GCN-based multi-view semi-supervised classification is heavily dependent on the structure of the graph neural network, we attempt to use attention mechanisms for forward propagation to reduce the over-dependence on the complex graph structure. Meanwhile, appropriate weights are assigned to multi-view data. In contrast to existing GCN methods, our approach focuses more on the contribution between features and views. The self-attention computation is independent and parallel, no additional matrix operations are required, and multiple output features can be parallelized on multiple nodes.

Table 1
Commonly used notations and their descriptions in this paper.

Notations	Descriptions
$\{\mathbf{X}^{(v)} \in \mathbb{R}^{N\times D_{(v)}}\}_{v=1}^{V}$	Input data of V views, N samples and $D_{(V)}$ features.
$\{\mathbf{Z}^{(v)} \in \mathbb{R}^{N\times D'_{(v)}}\}_{v=1}^{V}$	The output of view-specific embedding of V views.
$\{\mathbf{W}^{(v)}\}_{v=1}^{V}$	View-specific graph attention network weight.
$\sigma(\cdot)$	A non-linear activation function.
$\{\mathbf{S}^{(v)}\}_{v=1}^{V}$	Weight of view-specific feature embedding module.
$\{e_{ij}^{(v)}\}_{v=1}^{V}$	Coefficient between nodes i and j of the vth view.
$\{\alpha_{ij}^{(v)}\}_{v=1}^{V}$	Normalized coefficient between nodes i and j of the vth view.
$\zeta(\cdot)$	Fusion function with self-attention mechanism.
$\{\beta^{(v)}\}_{v=1}^{V}$	Feature fusion weight of graph embedding fusion.
$\mathbf{r}$	Shared weight vector for graph embedding fusion.
$\mathbf{Z}$	The output of graph embedding fusion module.

To obtain the high-level features, a linear transformation is applied to each node by a weight matrix $\mathbf{W} \in \mathbb{R}^{D_{(v)}\times D'_{(v)}}$. The coefficient of a self-attention mechanism for the vth view is formulated as

$$e_{ij}^{(v)} = \mathbf{S}^{(v)}\sigma\left(\left[\mathbf{W}^{(v)}\mathbf{x}_i^{(v)} \| \mathbf{W}^{(v)}\mathbf{x}_j^{(v)}\right]\right), \tag{5}$$

where $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ is a shared coefficient matrix of attention, and $\sigma(\cdot)$ is set to LeakyReLU as an activation function. Eq. (5) shows the impact of node i on node j of the vth view feature. Note that node i is associated with its neighbors and itself. Therefore, we compute $e_{ij}^{(v)}$ for each node, as a neighborhood coefficient of node i and node j. Then its normalization can be obtained using the softmax function. The normalized coefficient of the attention layer above is rewritten as

$$\alpha_{ij}^{(v)} = softmax(e_{ij}^{(v)}) = \frac{exp(e_{ij}^{(v)})}{\sum_{k\in\mathcal{N}_i} exp(e_{ik}^{(v)})}, \tag{6}$$

where $\mathcal{N}_i$ denotes the first-order neighbors of node i in the graph. We obtain the normalized attention coefficient, which can be used to calculate the linear combination of the corresponding multi-view features. To promote the learning process of self-attention, we extend our mechanism by adding multi-head attention in the first layer. For this goal, all learned embedding features are concatenated as the following form

$$\mathbf{z}_i^{(v)} = \|_{k=1}^{K}\xi\Big(\sum_{j\in\mathcal{N}_i}(\alpha_{ij}^{(v)})^k(\mathbf{W}^{(v)})^k\mathbf{x}_j^{(v)}\Big), \tag{7}$$

where $\|$ denotes the concatenation operator and K represents the number of layers in the network. Here, we use a nonlinear function ξ to construct the ultimate output of all nodes. Note that, if we apply Eq. (7) to the last layer, the final output would be influenced by all the afore layers, which may lead to undesirable results. For effectively optimizing the final output, we take the average of K layers for multi-view semi-supervised classification problems as

$$\mathbf{z}_i^{(v)} = \xi\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}(\alpha_{ij}^{(v)})^k(\mathbf{W}^{(v)})^k\mathbf{x}_j^{(v)}\Big). \tag{8}$$

At this point, we have obtained the consistent latent feature representation of the vth view data.

3.1.2. Graph embedding fusion module

Based on the consistent latent representation of each view obtained above, we focus on feature fusion in adaptive self-attention weights. Based on multi-view data to construct a graph structure, using the self-attention mechanism to capture multiple relationships between views, the global semantic information of the constructed graph can be captured during execution, and consistency information between different views can be extracted. The formulation of general approach fusion is given by

$$\mathbf{Z} = \zeta([\beta^{(1)}\mathbf{Z}^{(1)}, \ldots, \beta^{(V)}\mathbf{Z}^{(V)}]), \tag{9}$$

where $\{\beta^{(v)}\}_{v=1}^{V}$ is the set of fusion weights, and $\zeta(\cdot)$ denotes the fusion function. We use an attention mechanism to indicate directions of multiple view fusion for the semi-supervised classification task.

Considering that the latent embedding of the vth view data is denoted by $\mathbf{Z}^{(v)}$, the fusion weight can be obtained as follows

$$q_i^{(v)} = \mathbf{r}^T \cdot \sigma(\mathbf{Q}^{(v)} \cdot (\mathbf{z}_i^{(v)})^T + \mathbf{b}^{(v)}), \tag{10}$$

where $\mathbf{r}$ denotes a shared attention vector, and $\sigma(\cdot)$ is an activation function such as $tanh(\cdot)$. $\mathbf{Q}^{(v)} \in \mathbb{R}^{D_{(v)}\times D'_{(v)}}$ is the weight matrix of a fully-connected layer, and $\mathbf{b}^{(v)} \in \mathbb{R}^{1\times D'_{(v)}}$ is the bias vector. Then, we use a softmax function to normalize the attention weight q_i as follows:

$$\beta_i^{(v)} = softmax(q_i^{(v)}) = \frac{exp(q_i^{(v)})}{\sum_{v=1}^{V} exp(q_i^{(v)})}. \tag{11}$$

Finally, for all the nodes, the learned weights are obtained from $\beta^{(v)}$, which is comprised of a diagonal matrix $diag(\beta_1^{(v)}, \beta_2^{(v)}, \ldots, \beta_N^{(v)})$. After learning all the features of nodes, we combine all the latent embeddings to obtain the final predictive label matrix $\hat{\mathbf{Y}}$ of embedding $\mathbf{Z}$ as follows:

$$\hat{\mathbf{Y}} = softmax\Big(\sum_{v=1}^{V}\beta^{(v)}\mathbf{Z}^{(v)}\Big). \tag{12}$$

3.2. Training loss

For a multi-view semi-supervised classification problem, we use a cross-entropy loss function to estimate the difference between the predicted samples and the labeled samples to measure the fitting degree, expressed as

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbf{Y}_{ij}\ ln\ \hat{\mathbf{Y}}_{ij}, \tag{13}$$

where $\mathbf{Y}_{ij}$ denotes the one-hot coding of the given ground truths, and $\hat{\mathbf{Y}}_{ij}$ indicates the probability that the ith latent representative sample belongs to the class j.

Gathering all the aforementioned details, the procedures of the proposed graph attention fusion network for multi-view semi-supervised classification are summarized in Algorithm 1.

4. Experiments

In this section, we evaluate the proposed framework on several real-world multi-view datasets in terms of the semi-supervised classification task. Two main evaluation metrics are used to assess the performance of the above framework, compared with nine prior state-of-the-art methods, and the results obtained from the experiments are illustrated and analyzed.

4.1. Experimental settings

In this section, we introduce eight real-world datasets, nine state-of-the-art compared methods, and their parameter settings.

Algorithm 1 Graph Attention Fusion Network for multi-view semi-supervised classification (GAF-Net)

Require: Multi-view data $\mathcal{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{V}$, labeled samples $\{\mathbf{x}_i^{(v)}\}_{i=1}^{n}$, learning rate lr, and network layer number L, number of head attentions $heads$, dropout rate $dropout$.
Ensure: Predicted labels $\widehat{\mathbf{Y}}$ of test samples.
1: Initialize adjacency matrices $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$ via Gaussian kernel based KNN;
2: Initialize parameter matrices $\{\mathbf{W}^{(v)}\}_{v=1}^{V}$, $\{\mathbf{Q}^{(v)}\}_{v=1}^{V}$, coefficient matrices $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ and shared attention vector $\mathbf{r}$;
3: **while** not convergent **do**
4: **for** $v = 1 \to V$ **do**
5: Compute $\{e_{ij}^{(v)}\}$ with Eq. (5);
6: Perform $softmax$ operation on $\{e_{ij}^{(v)}\}$ to obtain $\{\alpha_{ij}^{(v)}\}$ with Eq. (6);
7: Calculate $\{\mathbf{z}_i^{(v)}\}$ with Eq. (8);
8: **end for**
9: Calculate the final label matrix $\widehat{\mathbf{Y}}$ with Eq. (12);
10: Update $\{\mathbf{W}^{(v)}\}_{v=1}^{V}$, $\{\mathbf{Q}^{(v)}\}_{v=1}^{V}$, $\{\mathbf{S}^{(v)}\}_{v=1}^{V}$ and $\mathbf{r}$ with backpropagation;
11: **end while**
12: **return** The predicted class label of the sample $\mathbf{x}_i$ is computed by $\widehat{\mathbf{y}}_i = \arg\max_j \widehat{\mathbf{Y}}_{ij}$ for any $i \in \{n+1, \cdots, N\}$.

4.1.1. Datasets

The proposed approach is applied on eight real-world multi-view datasets with various data categories, samples, views, and features.

3Sources[1] is a text dataset based on the news in three languages and consists of 169 news items by 6 topics, namely entertainment, politics, business, sports, health, and technology.

Animals is a dataset which consists of 30,475 animal pictures. Based on this, we generate a subset of 50 categories and 10,158 samples. Two types of features are extracted from the original data with two views: 4.096-D DECAF and 4,096-D VGG19.

Caltech20[2] is a subset of Caltech101, with 20 categories and 2386 images. We extract 6 views of features, including Gabor, WM, CENTRIST, HOG, GIST, and LBP.

HW[3] is composed of 2000 handwritten digital images of 10 categories, and each category contains 200 samples. Each image comes with 6 categories of related features.

MNIST[4] is a handwritten dataset with a total of 2000 samples, including 10 categories ranging from '0'–'9' with 3 views, where its features stand for IsoProjection, linear descriptive analysis, and neighborhood preserving embedding.

NUS-WIDE[5] is an image set, containing 2400 samples with 12 classes. Each sample has six view features, including 64 color histograms, 144 color correlograms, 73 edge direction histograms, 128 wavelet textures, 225 block-wise color moments, and 500 SIFT descriptors.

Scene15[6] contains both indoor and outdoor environments, including 15 scene categories and 4485 images.

WebKB-cornell[7] is a subset of WebKB about web pages and hyperlink data of Cornell University. There are 5 classes and 2 views with a total of 195 samples.

Table 2 shows the summary statistics of these datasets, reporting the numbers of views, features, and classes.

[1] http://mlg.ucd.ie/datasets/3sources.html
[2] https://data.caltech.edu/records/mzrjq-6wc02
[3] https://cs.nyu.edu/roweis/data.html
[4] http://yann.lecun.com/exdb/mnist/
[5] https://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[6] https://doi.org/10.6084/m9.figshare.7007177.v1
[7] http://www.cs.cmu.edu/webkb/

4.1.2. Comparison methods

KNN: K-Nearest Neighbors is a simple yet classical classification algorithm, where the class of the current node is determined via the nearest neighbors.

AMGL (Nie et al., 2016): Parameter-free auto-weighted multiple graph learning is a framework that can learn the weight automatically for semi-supervised classification and multi-view clustering tasks.

WREG (Yang et al., 2019): WREG is a supervised learning method to fuse multi-view data by mapping original features onto a low-dimensional subspace. The way to adaptively assign learned weights can maximize the correlative and complementary information for the classification task.

HLR-M^{2}VS (Xie et al., 2020): Hyper-Laplacian regularized multilinear multi-view self-representation model utilizes a unified view-specific feature space and an evidence-based tensor space to learn the global relevance and local structure among views to solve the semi-supervised classification task.

Co-GCN (Li et al., 2020): Co-GCN unifies three methods into one framework for the semi-supervised classification task, including co-training, spectral graph information, and neural network. The method can learn the spectral information from the rest views by combinatorial Laplacian to utilize the graph information.

AME-MSC (Wang, Wang & Guo, 2021): AME-MSC proposes an automatically learnable weight manifold embedding model for classifying unlabeled data using the category information of labeled data, which is experimentally proven to have positive robustness and generalization ability.

DSRL (Wang, Chen et al., 2022): DSRL utilizes a learnable sparse regularizer composed of multiple reusable blocks, where each block consists of a learnable piecewise linear activation function, enabling end-to-end multi-view clustering and semi-supervised classification tasks.

LGCN-FF (Chen et al., 2023): LGCN-FF divides a multi-step optimization strategy into some sub-problems by exploring the feature fusion network and learnable graph convolution network.

IMvGCN (Wu et al., 2023): IMvGCN provides an end-to-end framework in an interpretable way, which introduces a series of theoretical derivations to capture the multi-view embedding from feature and topology perspectives.

These comparison methods all serve the semi-supervised classification task. Notably, AMGL, Co-GCN, and AME-MSC can obtain the weight automatically through graph-based learning. However, our proposed method pays more attention to the features that contribute more to the consistency representation. Simultaneously, it does not discard the lesser contributive features completely.

4.1.3. Parameter setting

In our experiments, we follow the original parameter settings of the compared methods. Specifically, it is worth noting that we have also made some special modifications for certain parameters to obtain better results as follows:

KNN: We randomly choose the number of neighbors in the given training set from 1 to 10 for more robust label prediction.

WREG: We apply the trade-off parameter $\lambda = 0.1$. We set the value of the parameter k provided in the paper for the existing datasets and $k = 1$ for the rest.

HLR-M^{2}VS: We set $\lambda_1 = 0.2$ and $\lambda_2 = 0.4$ for the weight factors. The hyperedge of k nearest neighbors is constructed with a fixed setting of $k = 5$.

Co-GCN: We use a 2-layer GCN framework and configure the learning rate of the gradient descent method to be 0.001.

AME-MSC: We set the probability transition step as $step = 5$ in the deep rank walk. The regularization parameter β is fixed as 1.

DSRL: We apply a 10-layer network architecture and tune the learning rate of the optimization method to be 0.05.

Table 2
Statistics of the multi-view datasets used for the experiments.

Datasets	Data Types	Classes	Views	Samples	Features
3Sources	Online News Texts	6	3	169	3,068/3,560/3,631
Animals	Animal Images	50	2	10,158	4,096/4,096
Caltech20	Object Images	20	6	2,386	40/48/254/512/928/1,984
HW	Handwritten Images	10	6	2,000	27/153/157/301/481/596
MNIST	Handwritten Images	10	3	2,000	9/30/30
NUS-WIDE	Natural Images	12	6	2,400	64/73/128/144/225/500
Scene15	Environments Images	15	3	4,485	1,180/1,240/1,800
WebKB-cornell	Web Pages	5	2	195	195/1,703

Table 3
Node classification results of all compared methods when 10% samples are labeled.

Datasets/Methods	KNN	AMGL	WREG	HLR-M^{2}VS	Co-GCN	AME-MSC	DSRL	LGCN-FF	IMvGCN	GAF-Net
3Sources	46.0 (8.6)	39.8 (7.9)	54.0 (0.5)	69.0 (0.4)	44.1 (0.2)	77.1 (2.9)	77.2 (3.9)	66.3 (1.5)	89.5 (0.6)	**90.1 (1.1)**
Animals	73.6 (0.6)	70.9 (0.4)	82.1 (0.3)	72.7 (0.5)	80.2 (1.2)	66.3 (0.2)	76.4 (0.5)	72.2 (6.0)	82.8 (0.5)	**83.0 (0.3)**
Caltech20	67.0 (0.4)	45.0 (3.0)	49.2 (1.7)	80.4 (0.1)	67.0 (8.1)	70.4 (0.8)	80.9 (1.7)	71.9 (1.2)	43.0 (0.9)	**81.6 (1.1)**
HW	82.5 (0.7)	88.5 (0.8)	89.1 (1.0)	86.3 (2.1)	89.0 (1.1)	70.7 (1.0)	90.3 (0.9)	50.6 (12.9)	95.1 (0.3)	**95.2 (0.6)**
MNIST	86.4 (1.4)	69.5 (1.4)	83.9 (1.4)	89.6 (6.4)	87.7 (0.0)	81.5 (0.9)	88.3 (0.7)	88.5 (1.1)	89.7 (0.4)	**91.1 (1.0)**
NUS-WIDE	31.8 (1.2)	28.7 (0.4)	26.7 (1.7)	24.3 (0.1)	24.3 (0.1)	40.4 (1.9)	42.7 (0.7)	25.7 (3.5)	33.4 (1.3)	**43.3 (2.3)**
Scene15	45.2 (0.9)	68.4 (0.6)	52.3 (1.5)	67.4 (1.3)	58.6 (1.0)	60.1 (0.7)	66.8 (0.8)	18.8 (1.7)	66.7(0.0)	**69.4 (0.5)**
WebKB-cornell	48.9 (3.9)	53.1 (5.7)	50.3 (1.3)	51.8 (0.7)	50.6 (0.3)	43.3 (6.8)	49.2 (3.7)	52.5 (5.0)	42.4(0.5)	**60.2 (2.0)**

Table 4
Node classification results of all compared methods when 15% samples are labeled.

Datasets/Methods	KNN	AMGL	WREG	HLR-M^{2}VS	Co-GCN	AME-MSC	DSRL	LGCN-FF	IMvGCN	GAF-Net
3Sources	37.9 (4.3)	39.4 (1.6)	68.2 (0.7)	73.8 (0.5)	70.1 (0.2)	81.2 (3.1)	85.4 (3.2)	63.3 (0.8)	90.6 (0.3)	**92.3 (1.1)**
Animals	75.7 (0.3)	74.2 (0.3)	82.5 (0.3)	75.1(0.4)	75.7 (0.4)	72.7 (0.1)	79.4 (0.2)	73.1 (0.6)	83.0 (0.5)	**83.2 (0.0)**
Caltech20	69.7 (1.4)	49.6 (2.2)	75.4 (0.1)	84.3 (0.2)	71.4 (9.1)	73.8 (1.3)	81.0 (0.9)	74.6 (0.6)	46.0 (2.1)	**84.5 (0.8)**
HW	85.2 (1.5)	89.9 (0.4)	90.1 (0.8)	88.8 (1.0)	90.1 (0.0)	73.7 (0.5)	91.4 (0.6)	61.3 (6.3)	95.9 (0.2)	**96.1 (0.6)**
MNIST	88.1 (1.7)	69.3 (1.4)	86.2 (0.8)	89.8 (3.5)	88.8 (0.0)	62.4 (1.7)	88.8 (0.5)	88.8 (1.2)	89.4(0.2)	**91.0 (1.3)**
NUS-WIDE	33.3 (1.4)	31.3 (0.1)	28.8 (0.1)	26.6 (0.0)	25.2 (0.1)	44.8 (0.3)	44.5 (0.4)	25.7 (2.3)	33.5 (0.1)	**46.1 (1.0)**
Scene15	48.0 (0.7)	68.3 (0.6)	57.2 (1.4)	67.9 (1.2)	60.0 (2.1)	65.7 (0.1)	66.9 (0.6)	34.1 (5.0)	67.9 (0.1)	**68.6 (0.3)**
WebKB-cornell	49.3 (3.2)	56.1 (5.1)	64.9 (3.5)	52.4 (0.6)	53.3 (0.4)	49.6 (1.8)	50.1 (3.9)	51.6 (9.4)	45.1(0.9)	**71.0 (4.6)**

LGCN-FF: The weight decay is 0.01, and the learning rate is set as 0.01 for the fully-connected network and learnable GCN, 0.001 for the autoencoders.

IMvGCN: The learning rate is fixed at 0.01. For all datasets, we set the hyperparameter $\lambda = 0.5$.

With regard to the proposed method, a 2-layer GCN with an attention mechanism is adopted. The number of hidden variables ranges in $\{8, 16, 64\}$, and the dropout rate is selected from $\{0.3, 0.4, 0.6\}$. The learning rate of the optimizer is fixed to be 0.005. We provide several regularization methods as options, in addition to standard regularization. There is also max–min normalization for sparse data and outliers. We set the weight decay to 0.0005. The activation function of the attention layer is set as LeakyReLU(·). The KNN method is applied to construct the adjacency matrix, where the neighbor number is fixed as 9. The maximum number of iterations is set to 1,000. The proposed GAF-Net framework is implemented by PyTorch and runs on a machine with an I7-10800H CPU@2.3 GHz NVIDIA RTX 3060 GPU and 32G RAM. In addition to the Animals dataset, the running environment is completed using the A100 computing units provided by Google.

4.2. Semi-supervised classification

For all methods, we randomly select 10%, 15%, and 20% samples as labeled data, respectively, as shown in Tables 3–5.

We report the node classification results with average accuracy and standard deviation. Under the supervision of the above three labeling ratios, the remaining unlabeled data are used to evaluate the prediction performance of the proposed model and to calculate the cross-entropy loss. All methods are run 6 times. The experimental results indicate the proposed method achieves state-of-the-art performance on almost all test datasets. Compared with other methods, a significant improvement is achieved on the Scene15 dataset when the labeled samples rate is 0.1, outperforming the second best by 1.44%. The following datasets 3Sources, Animals, Caltech20, HW, MNIST, NUS-WIDE, and WebKB-cornell have yielded suboptimal results. The best performance is achieved on WebKB-cornell when the labeled ratio is 0.15, followed by the second-best performance on 3Sources and NUS-WIDE. When the labeled ratio is set as 0.2, increasing the number of labeled samples does not result in a significant improvement in overall accuracy. Instead, there is only a slight improvement observed across most datasets.

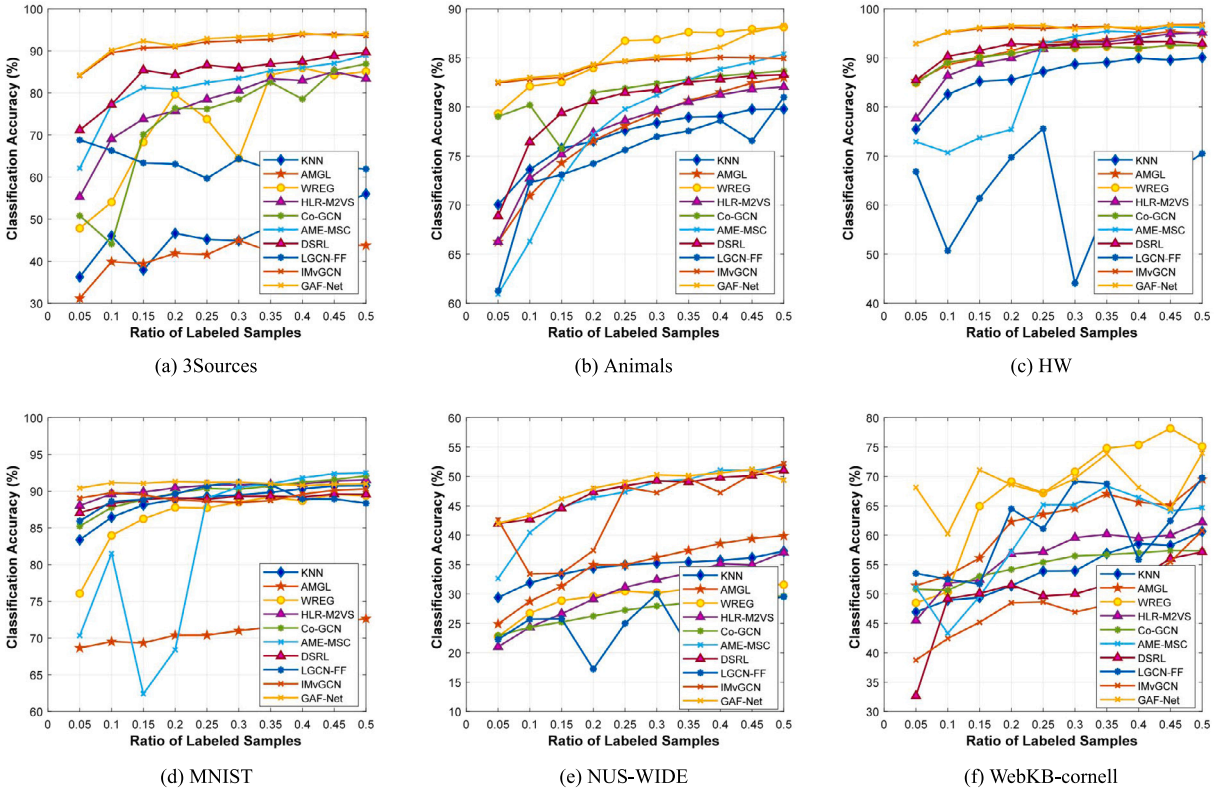
Fig. 2 shows the accuracy performance of nine compared methods under different ratios of labeled samples. This experimental result shows that the proposed method GAF-Net is particularly effective for datasets with a small amount of labeled samples. That is, GAF-Net performs well at less than 20% supervision rate and is particularly suitable for semi-supervised classification tasks.

4.3. Convergence analysis

Figs. 3–5 show the convergence of the loss function of GAF-Net. From the figures, we can obtain the following enlightening observations. We choose three representative datasets for an illustration, including Caltech20, HW, and MNIST. First, the loss typically decreases rapidly within 100 iterations for different supervision rates, because the loss is usually an approximately vertical line at the beginning of training. Then, after 100 iterations, there is a gradual stable value on the datasets except for the Caltech20 dataset when the supervision rate of 0.2 shows a significant loss of around 800 iterations. Second, the loss is significantly reduced after training on different datasets for a specific time, after about 600 iterations on the Caltech20 dataset, 200 iterations on the HW dataset, and 500 iterations on the MNIST dataset. The reason

Table 5
Node classification results of all compared methods when 20% samples are labeled.

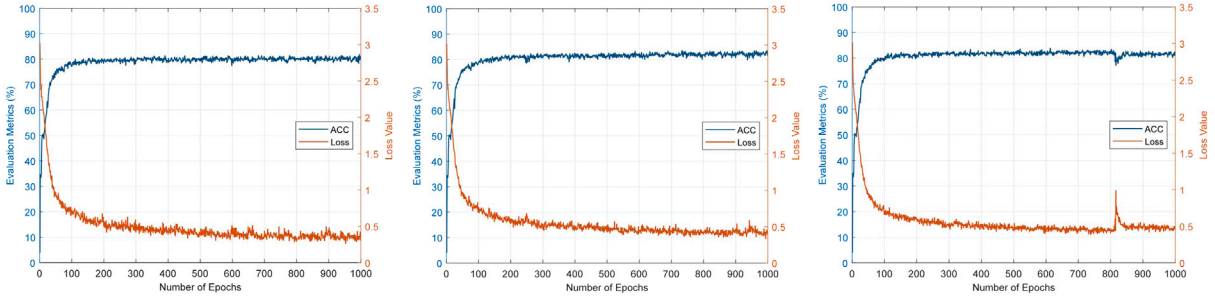
Datasets/Methods	KNN	AMGL	WREG	HLR-M^{2}VS	Co-GCN	AME-MSC	DSRL	LGCN-FF	IMvGCN	GAF-Net
3Sources	46.6 (9.3)	41.8 (2.3)	79.6 (0.8)	75.7 (0.2)	76.3 (0.3)	80.8 (0.9)	84.2 (2.9)	63.0 (0.2)	90.9(0.3)	**91.1 (2.1)**
Animals	76.5 (0.4)	76.5 (0.3)	83.9(0.4)	77.3(0.6)	81.4 (1.5)	77.1 (0.7)	80.6 (0.1)	74.2 (2.0)	84.2 (0.1)	**84.3 (1.0)**
Caltech20	70.5 (1.0)	83.2 (0.8)	75.0 (0.7)	**86.1 (1.1)**	74.8 (9.8)	83.0 (1.0)	82.8 (0.8)	74.3 (0.8)	47.2 (2.0)	84.2 (0.9)
HW	85.5 (0.4)	91.4 (0.5)	90.9 (0.2)	89.9 (0.0)	90.9 (0.0)	75.4 (1.2)	92.9 (0.5)	69.7 (7.5)	96.1(0.6)	**96.5 (0.5)**
MNIST	88.8 (0.6)	70.3 (1.1)	87.7 (0.9)	90.4 (0.0)	89.7 (0.0)	68.4 (2.5)	89.1 (0.5)	89.6 (1.7)	88.7 (0.4)	**91.3 (0.5)**
NUS-WIDE	34.4 (0.9)	34.9 (0.5)	29.5 (0.8)	29.1 (0.1)	26.2 (0.2)	46.3 (0.1)	47.3 (0.5)	17.2 (5.8)	37.4 (1.5)	**47.9 (0.9)**
Scene15	49.2 (0.5)	68.9 (0.5)	59.2 (1.3)	68.4 (0.6)	61.6 (3.4)	68.6 (0.7)	67.2 (0.4)	19.4 (0.8)	68.4 (0.1)	**69.1 (0.7)**
WebKB-cornell	51.3 (5.5)	62.3 (2.0)	69.1 (5.2)	56.7 (0.7)	54.1 (0.4)	57.2 (1.6)	51.5 (3.7)	64.4 (2.0)	48.5 (1.0)	**69.8 (2.9)**



(a) 3Sources (b) Animals (c) HW

(d) MNIST (e) NUS-WIDE (f) WebKB-cornell

Fig. 2. Classification performance of all compared methods for the labeled ratios ranging in {0.05, 0.10, ···, 0.50}.

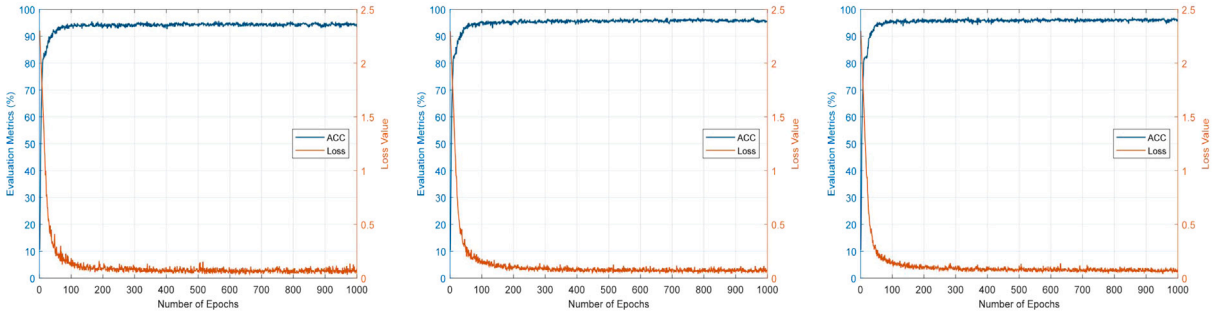


(a) Caltech20 (ratio = 0.10) (b) Caltech20 (ratio = 0.15) (c) Caltech20 (ratio = 0.20)

Fig. 3. Convergence curves of the proposed method on the Caltech20 dataset when the supervised ratio is fixed in {0.1, 0.15, 0.2}.

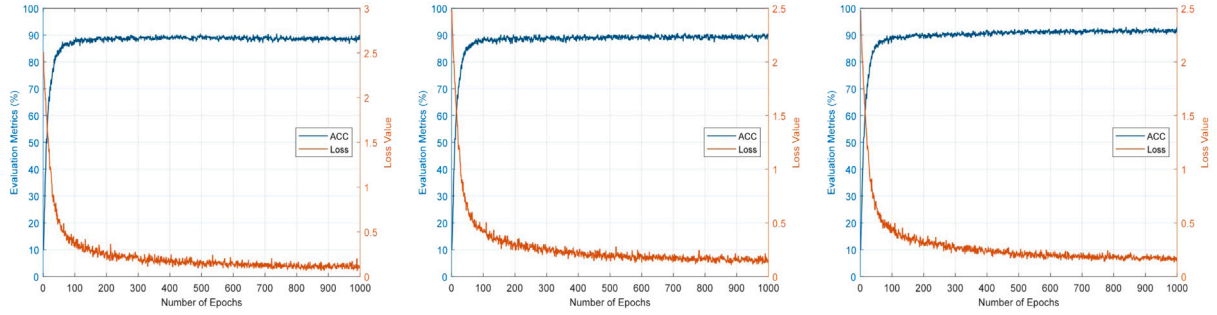
is that before learning a relatively stable embedding representation using the attention layer, the feature fusion network is also fusing the weights of multiple views to find a stable output when the loss value is large. Finally, as the loss continues to converge to a stable value, the accuracy of the corresponding unlabeled data gradually rises to a stable high value. Later in the training, it can be seen that the loss and accuracy also fluctuate somewhat after convergence, indicating that a stopping strategy with fewer iterations can be used, and experiments show that the average accuracy performs better.

We select all datasets and empirically demonstrate that our framework has better robustness under different hidden layers, as shown in Fig. 6. We set the hidden layers in the range of $\{8, 16, \ldots, 80\}$. While the ratio of labeled samples is fixed as 0.1, Fig. 6(a) demonstrates that for most datasets the classification accuracy fluctuates only slightly as the number of hidden layers increases. The same phenomenon is observed



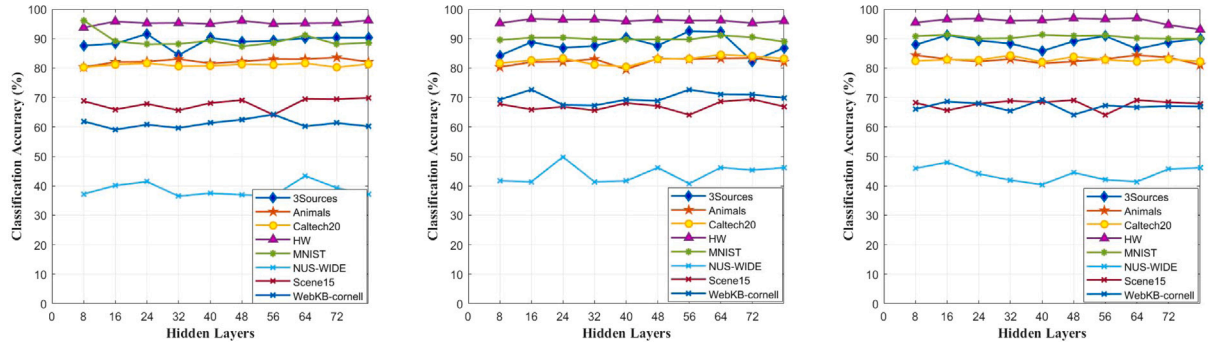
(a) HW (ratio = 0.1) (b) HW (ratio = 0.15) (c) HW (ratio = 0.2)

Fig. 4. Convergence curves of the proposed method on the HW dataset when the supervised ratio is fixed in {0.1, 0.15, 0.2}.



(a) MNIST (ratio = 0.1) (b) MNIST (ratio = 0.15) (c) MNIST (ratio = 0.2)

Fig. 5. Convergence curves of the proposed method on the MNIST dataset when the supervised ratio is fixed in {0.1, 0.15, 0.2}.



(a) Ratio of Labeled Samples = 0.1 (b) Ratio of Labeled Samples = 0.15 (c) Ratio of Labeled Samples = 0.2

Fig. 6. Accuracy curves under different hidden layers when the labeled ratio ranges in {0.1, 0.15, 0.2}.

in Fig. 6(b). Although, Fig. 6(c) has a slightly larger fluctuation than the two above. However, the overall trend of these three figures shows that the classification accuracy is not significantly affected by the increase in hidden layers. Even, individual datasets have the problem that the classification accuracy decreases as the number of hidden layers increases. The reason for this situation can be reduced to the overfitting of the network parameters. The reward is more pronounced when the supervision sample is small. Thus, we can use fewer hidden layers to achieve satisfactory classification performance.

4.4. Ablation study

In this subsection, we perform an ablation study to show the contribution of model components. It is noted that the proposed method GAF-Net is composed of the fusion of multiple views by the optimized attention weights. For a comparative approach to multi-view fusion, a weighted average method can be used. The main difference between this weighted average and the proposed version of fusion is that each view feature is treated equally and each view has the same contribution to the fused consistent representation. More specifically, for the fusion of multiple views, the weights of each view are not considered and a plain weighted average method is employed.

We construct a series of fusion experiments by using the average fusion $\mathcal{A}$, and the graph attention fusion $\mathcal{T}$, respectively. The impact of each component on the performance of the proposed method is shown in Table 6. It can be observed that the classification accuracy obtained by our method achieves the best results. The experimental results demonstrate that for different semi-supervised labeled ratios, the proposed method outperforms the results that only extract features and fuse them by a weighted average.

5. Conclusion

In this paper, we proposed a graph neural network framework based on an attention mechanism, which solved the multi-view classification problem by learning the most essential representations and constructing feature fusion networks. In the feature fusion networks,

Table 6
Ablation experiments of the proposed method under label ratios in $\{0.10, 0.15, 0.20\}$.

$\mathcal{A}$	$\mathcal{T}$	labeled ratio	3Sources	Animals	Caltech20	HW	MNIST	NUS-WIDE	Scene15	WebKB-cornell
✓		0.10	80.6	81.5	79.7	94.3	88.1	30.1	67.3	59.6
	✓		**90.1**	**83.0**	**81.6**	**95.2**	**91.1**	**43.3**	**69.4**	**60.2**
✓		0.15	82.0	82.0	81.1	96.0	90.0	32.9	68.2	64.5
	✓		**92.3**	**83.2**	**84.5**	**96.1**	**91.0**	**46.1**	**68.6**	**71.0**
✓		0.20	80.1	83.4	81.6	96.3	91.0	33.2	68.7	64.2
	✓		**91.1**	**84.3**	**84.2**	**96.5**	**91.3**	**47.9**	**69.1**	**69.8**

attention mechanisms were used to extract the importance of different views and to fuse different consistent representations by constructing attentional networks and exploiting complementarity. The graph fusion process was performed by an attention layer that adaptively fused multiple topological graphs from multiple views. Finally, the experimental results also demonstrated that the proposed framework achieved promising performance in multi-view semi-supervised classification tasks.

There are still several research directions to be further explored. First, although GCN-based approaches achieve satisfactory results for node classification tasks, the construction of multiple views in almost all models is developed based on a fixed adjacency matrix, resulting in limited model expressiveness. Second, existing GCN methods usually lack interpretability. By using an attention-based mechanism, nodes in the same neighborhood can be implicitly assigned to different weights. Therefore, it is an interesting research direction to explore the interpretability of the model by analyzing the learned attention weights. Finally, it is quite common to obtain multi-view data with missing data from a certain view or some views due to sensors, etc. Therefore, semi-supervised multi-view classification for such incomplete views may be also a worthwhile research direction.

CRediT authorship contribution statement

Na Song: Conceptualization, Methodology, Software, Writing – original draft. **Shide Du:** Software, Visualization, Validation, Writing – review & editing. **Zhihao Wu:** Investigation, Validation, Writing. **Luying Zhong:** Investigation, Validation. **Laurence T. Yang:** Revision, Supervision. **Jing Yang:** Revision, Proofread. **Shiping Wang:** Conceptualization, Methodology, Validation, Supervision.

Declaration of competing interest

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is in part supported by the Development Program of China under Grant 2022ZD0118300, the National Natural Science Foundation of China under Grants U21A20472, 62276065, 62302131, 62302130, 62276146, the National Key Research and Development Plan of China under Grant 2021YFB3600503, and the Science and Technology Project of Putian under Grant 2022GZ2001ptxy11.

References

Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, *6*, Article 100134.

Chen, L., Cao, J., Wang, Y., Liang, W., & Zhu, G. (2022). Multi-view graph attention network for travel recommendation. *Expert Systems with Applications*, *191*, Article 116234.

Chen, C., Fragonara, L. Z., & Tsourdos, A. (2021). GAPointNet: Graph attention based point neural network for exploiting local feature of point cloud. *Neurocomputing*, *438*, 122–132.

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

Chen, M., Huang, L., Wang, C., Huang, D., & Lai, J. (2021). Relaxed multi-view clustering in latent embedding space. *Information Fusion*, *68*, 8–21.

Chen, M., Liu, T., Wang, C., Huang, D., & Lai, J. (2022). Adaptively-weighted integral space for fast multiview clustering. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 3774–3782).

Chen, Z., & Wang, S. (2022). A review on matrix completion for recommender systems. *Knowledge and Information Systems*, *64*, 1–34.

Chen, L., Wu, L., Hong, R., Zhang, K., & Wang, M. (2020). Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the thirty-fourth AAAI conference on artificial intelligence* (pp. 27–34).

Chen, Z., Zhao, W., & Wang, S. (2021). Kernel meets recommender systems: A multi-kernel interpolation for matrix completion. *Expert Systems with Applications*, *168*, Article 114436.

Deng, Z., Huang, L., Wang, C., Lai, J., & Yu, P. S. (2019). Deepcf: A unified framework of representation learning and matching function learning in recommender system. In *Proceedings of the thirty-third AAAI conference on artificial intelligence* (pp. 61–68).

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proceedings of conference on computer vision and pattern recognition workshop* (pp. 178–178).

Fu, L., Chen, Z., Chen, Y., & Wang, S. (2022). Unified low-rank tensor learning and spectral embedding for multi-view subspace clustering. *IEEE Transactions on Multimedia*, 1–14. http://dx.doi.org/10.1109/TMM.2022.3185886.

Gan, M., & Ma, Y. (2022). DeepInteract: Multi-view features interactive learning for sequential recommendation. *Expert Systems with Applications*, *204*, Article 117305.

Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., & Shen, C. (2021). Graph attention tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9543–9552).

Han, X., Ren, Z., Zou, C., & You, X. (2022). Incomplete multi-view subspace clustering based on missing-sample recovering and structural information learning. *Expert Systems with Applications*, *208*, Article 118165.

Han, Z., Yang, F., Huang, J., Zhang, C., & Yao, J. (2022). Multimodal dynamics: Dynamical fusion for trustworthy multimodal classification. In *Proceedings of conference on computer vision and pattern recognition* (pp. 20675–20685).

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*, 261–266.

Huang, S., Zhang, Y., Fu, L., & Wang, S. (2022). Learnable multi-view matrix factorization with graph embedding and flexible loss. *IEEE Transactions on Multimedia*, http://dx.doi.org/10.1109/TMM.2022.3157997.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th international conference on learning representations* (pp. 24–26).

Li, S., Li, W.-T., & Wang, W. (2020). Co-GCN for multi-view semi-supervised learning. In *Proceedings of the thirty-fourth AAAI conference on artificial intelligence* (pp. 4691–4698).

Lin, Z., Kang, Z., Zhang, L., & Tian, L. (2023). Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, *35*, 1872–1880.

Liu, B., Zhong, H., & Xiao, Y. (2022). New multi-view classification method with uncertain data. *ACM Transactions on Knowledge Discovery from Data*, *16*, 19:1–19:23.

Musetti, A., Manari, T., Billieux, J., Starcevic, V., & Schimmenti, A. (2022). Problematic social networking sites use and attachment: A systematic review. *Computers in Human Behavior*, Article 107199.

Nie, F., Li, J., & Li, X. (2016). Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 1881–1887).

Nie, L., Qu, L., Meng, D., Zhang, M., Tian, Q., & Bimbo, A. D. (2022). Search-oriented micro-video captioning. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 3234–3243).

Nie, L., Song, X., & Chua, T.-S. (2022). *Learning from multiple social networks*. Springer Nature.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2021). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, *32*, 604–624.

Pan, E., & Kang, Z. (2021). Multi-view contrastive graph clustering. *Advances in Neural Information Processing Systems*, *34*, 2148–2159.

Tang, C., Liu, X., Zheng, X., Li, W., Xiong, J., Wang, L., Zomaya, A. Y., & Longo, A. (2022). DeFusionNET: Defocus blur detection via recurrently fusing and refining discriminative multi-scale deep features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*, 955–968.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems* (pp. 5998–6008).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

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

Wang, X., Fu, L., Zhang, Y., Wang, Y., & Li, Z. (2022). Mmatch: Semi-supervised discriminative representation learning for multi-view classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*, 6425–6436.

Wang, Z., Shen, Z., Zou, H., Zhong, P., & Chen, Y. (2022). Retargeted multi-view classification via structured sparse learning. *Signal Processing*, *197*, Article 108538.

Wang, S., Wang, Z., & Guo, W. (2021). Accelerated manifold embedding for multi-view semi-supervised classification. *Information Sciences*, *562*, 438–451.

Wang, S., Wang, Z., Lim, K., Xiao, G., & Guo, W. (2021). Seeded random walk for multi-view semi-supervised classification. *Knowledge-Based Systems*, *222*, Article 107016.

Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., & Pei, J. (2020). Am-GCN: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1243–1253).

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

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *Proceedings of the international conference on machine learning* (pp. 6861–6871).

Xia, W., Wang, Q., Gao, Q., Zhang, X., & Gao, X. (2021). Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, *24*, 3182–3192.

Xia, W., Wang, S., Yang, M., Gao, Q., Han, J., & Gao, X. (2022). Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation. *Neural Networks*, *145*, 1–9.

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

Xu, C., Dai, Y., Lin, R., & Wang, S. (2020). Deep clustering by maximizing mutual information in variational auto-encoder. *Knowledge-Based Systems*, *205*, Article 106260.

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

Zhang, C., Cui, Y., Han, Z., Zhou, J. T., Fu, H., & Hu, Q. (2022). Deep partial multi-view learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*, 2402–2415.

Zhang, Y., Guo, X., Ren, H., & Li, L. (2021). Multi-view classification with semi-supervised learning for SAR target recognition. *Signal Processing*, *183*, Article 108030.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

Zhao, L., Wang, X., Liu, Z., Yuan, H., Zhao, J., & Zhou, S. (2023). Deep probability multi-view feature learning for data clustering. *Expert Systems with Applications*, *217*, Article 119458.

Zt, A., Yw, B., Xiang, W. C., Xh, D., Xh, A., & Tsc, C. (2020). MGAT: Multimodal graph attention network for recommendation. *Information Processing & Management*, *57*, Article 102277.