Multi-Label Graph Convolutional Network Representation Learning

Min Shi, *Student Member, IEEE,* Yufei Tang, *Member, IEEE,* Xingquan Zhu, *Senior Member, IEEE,* and Jianxun Liu

Abstract—Knowledge representation of networked systems is fundamental in many disciplines. To date, existing methods for representation learning primarily focus on networks with simplex labels, yet real-world objects (nodes) are inherently complex in nature and often contain rich semantics or labels. For example, a user may belong to diverse interest groups of a social network, resulting in multi-label networks for many applications. A multi-label network not only has multiple labels for each node, the labels are often highly correlated making existing methods ineffective or even fail to handle such correlation for node representation learning. In this paper, we propose a novel multi-label graph convolutional network (MuLGCN) for learning node representation. To fully explore label-label correlation and network topology structures, we propose to model a multi-label network as two Siamese GCNs: a node-node-label graph and a label-label-node graph. The two GCNs each handle one aspect of representation learning for nodes and labels, respectively, and are seamlessly integrated in one objective function. The learned label representations can effectively preserve the intra-label interaction and node label properties, and are aggregated to enhance the node representation learning under a unified training framework. Experiments and comparisons on multi-label node classification validate the effectiveness of our proposed approach.

Index Terms—Multi-label graph, graph learning, network embedding, network representation learning, multi-label learning, neural networks

1 Introduction

Graphs have become increasingly common structures for organizing data in complex systems, such as sensor networks, citation networks, social networks, and many more [1]. The advancement raises new requirement of efficient network representation or embedding learning algorithms for various real-world applications, which seek to learn low-dimensional vector representations of all nodes preserving graph topology structures, such as edge links, degrees, and communities *etc*. The graph edges inherently reflect semantic relevance between nodes, where nodes with similar neighborhood structures tend to share identical labeling information, *i.e.*, forming clusters characterized by a single grouping label. For examples, in a scientific collaboration network, two connected authors often belong to a common area of science [2], [3], and in a protein-protein interaction network, proteins co-appearing in identical protein complexes are likely to have similar biological functions.

To date, a large body of work has been focused on the representation learning of graphs with simplex labels [4], [5], where each node only has one single label. The node labels are used to implicitly model node relationships, *i.e.*, two nodes in a neighborhood are enforced to have similar labels in the learning process. In reality, graph nodes associated with multiple labels are ubiquitous in many real-world applications. For example, a photograph, in an image network, can belong to more than one semantic class, such as *sunsets* and *beaches*. Due to disease comorbidities, in a patient network, a patient may suffer from *diabetes* and *kidney cancer* at the same time. Similarly, in many social networks, such as BlogCatalog and Flickr, users are allowed to join multiple groups that respectively represent their diverse interests. For all these networks, each node not only has content (or features), it is also associated with multiple class labels.

In general, multi-label graphs primarily differ from simplex-label graphs in twofold. First, every node in a multi-label graph could be associated with a set of labels. As a result, graph structures usually encode much more complicated relationships between nodes sharing similar labels, *i.e.*, an edge could either reflect a simple relationship between single labels or interpret a very complex relationship between multiple combined labels. Second, it has been widely accepted that label correlations and dependencies are widespread between multiple labels [6], [7], *i.e.*, the *sunsets* are frequently correlated with the *beaches*, and *diabetes* could finally lead to *kidney cancer*. Therefore, the correlation and interaction between labels could provide implicit and supplemental factors to enhance and differentiate node relationships that cannot be explicitly captured by the discrete and independent labels in a simplex-label graph.

Indeed, multi-label learning is a fundamental problem in the machine learning community [8], with significant attentions in many research domains such as computer vision [9], text classification [10], and tag recommendation

This work was supported in part by the U.S. National Science Foundation through Grant Nos. IIS-1763452, CNS-1828181, OAC-2017597, & IIS-2027339, and an Early-Career Research Fellowship from the Gulf Research Program (GRP) of the National Academies of Sciences, Engineering, and Medicine (NASEM).

- *Min Shi, Yufei Tang and Xingquan Zhu are with the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, USA.*
  E-mail: mshi2018@fau.edu, tangy@fau.edu, xzhu3@fau.edu
- *Jianxun Liu is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Hunan, China.*
  E-mail: ljx529@gmail.com.

Manuscript received Jan. 02, 2020; revised XXX XXX, 2020.

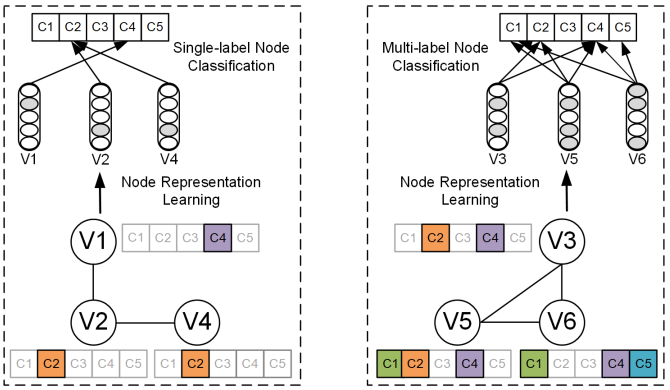


Fig. 1: Illustration of the difference between simplex-label graph learning *vs.* multi-label graph learning, where the labels of each node are highlighted (using different colors). In a simplex-label graph, each node is associated with only one label. In a multi-label graph, each node may be associated with multiple labels and these node labels are often highly correlated to represent node semantics.

[11]. However, research on multi-label graph learning is still in its infancy. Existing methods either consider graphs with simplex labels [5], [12] or treat multiple labels as plain attribute information to enhance the graph learning process [13], [14]. Such learning paradigms, however, overlook the fact that the information of one label may be helpful for the learning of another related label [6]— the label correlations may provide helpful extra information especially when some labels have insufficient training examples. To address this constraint and meanwhile advance the graph learning theory, we propose a multi-label graph representation learning framework in this paper, where each node has a collection of features as well as a set of labels. Figure 1 illustrates the difference between our studied problem and the traditional simplex-label graph learning. The key for multi-label graph learning is to combine network structures, node features, and label correlations for enriched node relationship modeling in a mutually reinforced manner.

Incorporating node labels and their correlations with topology structures for graph representation learning is a nontrivial task. First, in a multi-label graph, two linked nodes may share one or multiple identical labels, thus their affinity cannot be simply determined by one observed edge that is indistinguishable from others. Second, while each label can be seen as an abstraction of nodes sharing similar network structures and features, the label-label correlations may significantly impact on the node-node interactions, thus it is hard to constrain and balance the two aspects of relation modeling for an optimal graph representation learning as a whole. Recently, a general class of neural network called Graph Convolutional Networks (GCN) [15] shows superb performance in learning node representations from graphs by performing supervised single-label node classification training. GCN operates directly on a graph and induces embedding vectors based on the spectral convolutional filter that enforces each node to aggregate features from all neighbors to form its representation.

In this paper, we advance the graph convolutional networks to multi-label node classification and propose a novel model called Multi-Label GCN (MuLGCN) to specifically handle the multi-label graph learning problem. MuLGCN contains two Siamese GCNs to learn label and node representations from a high-layer label-label-node graph and a bottom-layer node-node-label graph, respectively. The top-layer graph learning serves to model label correlations, which only updates the label representations with preserved labels, label correlations and node community information by performing a single-label classification. The derived label representations are subsequently aggregated to enhance the bottom-layer graph learning, which carries out node representation learning from graph structures and features by performing a multi-label classification. Learning in the two layers can enhance each other in an alternative training manner to optimize a collective classification objective.

It is worth noting that a recent work, ML-GCN, proposes to build a directed graph over multi-label objects labels, and uses the constructed label-label correlation network to learn label embedding (using two-layer GCNs) [16]. While both our approach (MuLGCN) and ML-GCN consider label-label correlation, using GCN based embedding learning, our method is essentially different from ML-GCN mainly because of the following four key aspects:

- **Specific vs. generic data domains:** ML-GCN focuses on the specific multi-label image classification task, where each object represents an image. In comparison, our method is able to handle generic objects, such as documents or simple feature represented instances. Our method can be applied to image domains, and can also be used for generic data classification.
- **I.I.D. vs. Networked samples:** ML-GCN focuses on independent and identically distributed (i.i.d.) image samples where image objects are independent and have no linkage relationships between objects, whereas our approach is able to handle networked objects, where objects share inter-dependency and linkage relationships.
- **Homogeneous vs. heterogeneous GCN relationships:** While both ML-GCN and our method propose solutions under a GCN framework, the underlying networks are different. ML-GCN derives label representations from the constructed label-label network. In other words, the network in ML-GCN is a simple homogeneous network (nodes are the same type) where each node is a label and linkage denotes label relationships. In comparison, our approach builds a heterogeneous network, including label nodes and instance nodes. The two types of nodes contribute useful information for discriminative representation learning through the interaction of label-label-node graph and node-node-label graph. Therefore, in our model both label correlations and node interactions could enhance each other for semantic enriched label and node representation learning.
- **Static vs. dynamic combination of representations:** ML-GCN adopts a simple dot-production mechanism to combine label representation and image features (learned from Convolutional Neural Network CNN) for unified image representation generation and classification at the end of the design. In comparison, our

approach allows dynamic semantic interaction of labels and nodes during the learning, which is more efficient to leverage label correlations for multi-label learning of graph structured data as demonstrated in the experimental comparison results.

To summarize, our main contribution is threefold:

1) We advance the traditional simplex-label graph learning to a multi-label graph learning setting, which is more general and common in many real-world graph-based systems.
2) Unlike many existing methods that treat multiple labels as flat attributes, we propose to leverage label correlations to strengthen and differentiate edge relationships between nodes.
3) The proposed model, MuLGCN, simultaneously integrates graph structures, features, and label correlations to enhance node representation learning and classification of multi-label graphs.

The rest of the paper is organized as follows. Sec. 2 reviews related work. Sect. 3 introduces some preliminaries, including definition of the multi-label graph learning problem and the graph convolutional networks used in our approach. The proposed model for multi-label graph embedding is introduced in Sect. 4, followed by experiments and comparisons in Sect. 5 and conclusion in Sec. 6.

2 Related Work

Our research is related to multi-label learning and graph representation learning.

2.1 Multi-label Learning

Multi-label learning is a well established research problem in the machine learning community with applications ranging from document classification and gene function prediction to automatic image annotation and video classification [17], [18]. In a multi-label learning task, each instance is associated with multiple labels represented by a sparse label vector. The objective is to learn a classifier that can automatically assign an instance with the most relevant subset of labels [8]. Techniques for multi-label classification learning can be broadly divided into two categories [19]: transformation-based or algorithm adaption-based. The former generally transforms the multi-label classification task into a series of binary classification problems [6], [20] and the latter tries to generalize some popular learning algorithms to enable a multi-label learning setting [21], [22].

Multi-label learning methods for graph-based data receive very little attention in the past. DeepWalk [23] was proposed to learn graph representations that are then used for training a multi-label classifier. However, DeepWalk only exploits graph structures, with valuable label and label correlation information not preserved in learned node embeddings. Wang et al. [4] and Huang et al. [14] proposed to leverage labeling information along with graph structures for enriched representation learning. However, these methods either consider simplex-label graphs or treat multiple labels as plain attribute features to support graph structure modeling. Such paradigms still neglect frequent label correlations and dependencies which are demonstrably helpful properties in multi-label learning problems [9], [10].

2.2 Graph Representation Learning

Graph representation learning [1], [24] seeks to learn low-dimensional feature vector representations of a given network, to directly benefit various downstream analytic tasks like link prediction and node classification. Traditional methods in this area are generally developed based on shallow neural models, such as DeepWalk [23], Node2vec [25], and LINE [26]. To preserve node neighborhood relationships, they typically perform truncated random walk over the whole graph to generate a collection of fixed-length node sequences, where nodes within the same sequences are assumed to have semantic connections and will be mapped to be close to each other in the learned embedding space. However, above methods only consider modeling the edge links to constrain node relations, which may be insufficient especially when the network structures are very sparse. To mitigate this issue, many methods [3], [27] are proposed to additionally embed the rich network contents or features associated such as the user profiles in a social network and the publication descriptions in a citation network. For example, TriDNR [28] was proposed to simultaneously learn from network structures and textual contents, where structures and texts are mutually boosted to collectively constrain the similarities between learned node representations. In general, most real-word graphs are sparse in connectivity (*e.g.*, each node only connects several others in the huge node space), while node contents or features can be leveraged to either enhance node relevance or repair the missing links over the original network structure [29].

The above representation learning methods belong to the class of shallow neural network models, which may have limitations in learning complex relational patterns between graph nodes. Recently, there is a growing interest in adapting deep neural networks to handle non-Euclidean graph data [12], [30]. Several works seek to apply the concepts of convolutional neural networks to process arbitrary graph structures [15], [26], with GCN [15] achieving state-of-the-art representation learning and node classification performance on a number of benchmark graph datasets. Following this success, Yao el al. [31] proposed a text GCN for document embedding and text classification based on a constructed heterogeneous word-document graph. Graph Attention Networks (GAN) [12] are another recently proposed end-to-end neural network structure similar to GCN, which introduce attention mechanism assigning large weight values to important nodes, walks, or models. Inspired by these deep neural models targeted at mostly the simplex-label graphs, our research generalizes GCN and proposes a novel training framework, MuLGCN, to address the multi-label graph learning problem.

3 Problem Definition & Preliminaries

3.1 Problem Definition

A multi-label graph is represented as $G = (\mathbf{v}, \mathbf{e}, \mathbf{c}, \mathbf{A}, \mathbf{U}, \mathbf{X})$, where $\mathbf{v} = \{v_i\}_{i=1,\cdots,n}$ is a set of unique nodes, $\mathbf{e} = \{e_{i,j}\}_{i,j=1,\cdots,n;\ i\neq j}$ is a set of edges and $\mathbf{c} = \{c_r\}_{r=1,\cdots,m}$ is a set of unique labels, respectively. n is the total number of nodes in the graph and m is the total number of unique labels in the labeling space. For clarification purpose, we refer to each

TABLE 1: Notations used in the proposed model.

Symbols	Description
$\mathbf{v}$	set of vertices (nodes)
$\mathbf{c}$	set of class labels of all nodes
n	total number of nodes $n = \|\mathbf{v}\|$
m	total number of labels $m = \|\mathbf{c}\|$
$\mathbf{X}$	feature matrix for all n nodes
$\mathbf{H}$	feature matrix for all m labels
$\mathbf{F}$	adjacency matrix for the label-label-node graph
$\tilde{\mathbf{F}}$	the normalized adjacency matrix of $\mathbf{F}$
$\mathbf{E}$	adjacency matrix for the node-node-label graph
$\tilde{\mathbf{E}}$	the normalized adjacency matrix of $\mathbf{E}$
$\mathbf{A}$	adjacency matrix for graph between common nodes
$\mathbf{B}$	adjacency matrix for graph between label nodes
d_f	the input feature size of label and common nodes
d_h	the hidden convolution embedding size
d_e	the output convolution embedding size
$\mathbf{O}^l$	the learned label node embeddings
$\mathbf{O}^v$	the learned common node embeddings
$\mathbf{D}_f, \mathbf{D}_c$	degree matrices
$\mathbf{D}_e, \mathbf{D}_a$	degree matrices
$\mathbf{I}_{m+n}$	identity matrix
$\mathbf{I}_m, \mathbf{I}_n$	identity matrices
$\mathbf{W}_0^l, \mathbf{W}_1^l$	learned parameters for label-label-node network
$\mathbf{W}_0^v, \mathbf{W}_1^v$	learned parameters for node-node-label network
$\mathbf{Y}^l$	one-hot label indicator matrix of all m label nodes
$\mathbf{Y}^v$	one-hot label indicator matrix of all n common nodes
$\mathbf{y}$	set of nodes that have labels for supervised training
E	total number of edges for the node-node-label graph
L	total number of edges for the label-label-node graph

node in $\mathbf{v}$ as a "common node". Likewise, each label c_r will also correspond to a "label node" in the node-node-label graph and label-label-node graph.

$\mathbf{A}$ is an $n \times n$ adjacency matrix with $\mathbf{A}_{i,j} = w_{i,j} > 0$ if $e_{i,j} \in \mathbf{e}$ and $\mathbf{A}_{i,j} = 0$ if $e_{i,j} \notin \mathbf{e}$. $\mathbf{U}$ is an $n \times m$ affiliation matrix of labels with $\mathbf{U}_{i,r} = 1$ if v_i has label $c_r \in \mathbf{c}$ or otherwise $\mathbf{U}_{i,r} = 0$. Finally, $\mathbf{X} \in \mathbb{R}^{n \times d_f}$ is a matrix containing all n nodes with their features, *i.e.*, $\mathbf{X}_{v_i} \in \mathbb{R}^{d_f}$ represents the feature vector of node v_i, where d_f is the feature vector's dimension.

In this paper, multi-label graph learning **aims** to represent nodes of graph G in a new d_e-dimensional feature space $\mathcal{H}^{d_e}$, embedding information of graph structures, features, labels and label correlations preserved, *i.e.*, learning a mapping $f : G \rightarrow \{\mathbf{h}_{v_i}\}_{i=1,\cdots,n}$ such that $\mathbf{h}_{v_i} \in \mathcal{H}^{d_e}$ can be used to accurately infer labels associated with node v_i.

3.2 Graph Convolutional Networks

GCN [24] is a general class of graph neural networks that operate directly on graphs for node representation learning by encoding both the graph structures and node features. In this paper, we focus on spectral-based GCN [17], which assumes that neighborhood nodes tend to have identical labels guaranteed by each node gathering features from all neighborhoods to form its representation. Given a network $G = (\mathbf{v}, \mathbf{e}, \mathbf{X})$, which has n nodes and each node has a set of d_f-dimensional features ($\mathbf{X} \in \mathbb{R}^{n \times d_f}$ denotes the feature vector matrix of all nodes), GCN takes this graph as input and obtains the new low-dimensional vector representations of all nodes though a convolutional learning process. More specifically, with one convolutional layer, GCN is able to preserve the 1-hop neighborhood relationships between nodes, where each node will be represented as a d_e-dimension vector. The output feature matrix for all nodes $\mathbf{X}^{(1)} \in \mathbb{R}^{n \times d_e}$ can be computed by:

$$\mathbf{X}^{(1)} = \rho(\tilde{\mathbf{A}}\mathbf{X}^{(0)}\mathbf{W}_0) \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{A})\mathbf{D}^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix. $\mathbf{D}$ is the degree matrix of $(\mathbf{I} + \mathbf{A})$ and $\mathbf{I}$ is an identity matrix with a corresponding shape. $\mathbf{X}^{(0)} \in \mathbb{R}^{n \times d_f}$ is the input feature matrix (*e.g.*, $\mathbf{X}^{(0)} = \mathbf{X}$) for GCN and $\mathbf{W}_0 \in \mathbb{R}^{d_f \times d_e}$ is the first convolutional layer weight matrix. ρ is an activation function such as the *ReLU* represented by $\rho(x) = \max(0, x)$. If it is necessary to encode k-hop neighborhood relationships, one can easily stack multiple GCN layers, where the output node features of the jth ($1 \leq j \leq k$) layer is calculated by:

$$\mathbf{X}^{(j+1)} = \rho(\tilde{\mathbf{A}}\mathbf{X}^{(j)}\mathbf{W}_j) \tag{2}$$

where $\mathbf{W}_j \in \mathbb{R}^{d_h \times d_h}$ is the weight matrix for the jth layer and d_h is the feature vector dimension output in the hidden convolutional layer.

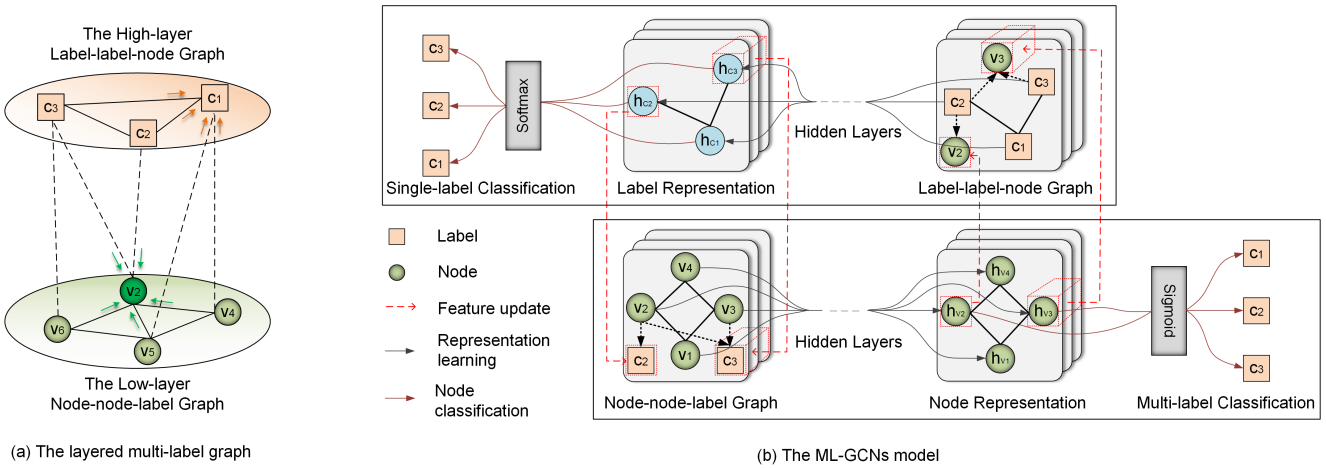
4 The Proposed Approach

In this section, we first present the proposed Multi-Label Graph Convolutional Networks (MuLGCN) model, where node representations are learned and trained through the supervised learning. Then, we provide the training and optimization details which incorporate node and label representation learning in a collective objective, followed by the computation complexity analysis of the MuLGCN.

4.1 Multi-Label Graph Convolutional Networks

As discussed in the previous sections, the key and challenge for multi-label graph learning are to simultaneously learn from the topology structures, node features, node labels, and label correlations, where different aspects of learning could enhance each other to achieve a global good network representation. To support the incorporation of labels, we can simply build a heterogeneous node-label graph similar to the text GCN [31], where common nodes and label nodes are directly connected by their labeling relationships. However, such a diagram makes it hard to model high-order label correlations since labels must reach each other through common nodes, *i.e.*, one cannot directly encode k-hop neighborhood node relations and label correlations by a GCN with k convolutional layers. To enable immediate and flexible label interactions, we consider a stratified graph structure shown in Fig. 2(a), which is defined as below:

Label-label-node graph: In this graph, each node denotes a lable c_r, and label nodes connect to each other according to their co-occurrence relationships, *i.e.*, two labels are connected (sharing an edge) if they have ever been used together to annotate graph nodes, *i.e.*, suppose any graph node $v_i \in \mathbf{v}$ contains both label c_1 and c_2, then the two labels c_1 and c_2 build an edge in the graph. Meanwhile, each label



(a) The layered multi-label graph

(b) The ML-GCNs model

Fig. 2: The proposed MuLGCN model for multi-label graph learning. (a) shows a multi-label graph organized in two layers. (b) shows the proposed architecture that contains two Siamese GCNs to learn from the label-label-node graph (top layer) and node-node-label graph (bottom layer), respectively. The top panel uses label-label-node graph to learn label representation (from right to left), and the bottom panel uses node-node-label graph to learn node representation (from left to right).

node c_r also points to a common node v_i, if c_r is one of the labels used to annotate v_i.

Node-node-label graph: In this graph, common nodes link together using their original graph topology structure. Meanwhile, each labeled common node v_i also points to a label node c_r if c_r is one of the labels used to annotate v_i.

For both graphs, by considering label nodes as attributes of common nodes or vice versa, the common nodes and label nodes are able to interact with each other during the learning, allowing label information, label correlation, and graph structures to be collectively encoded for optimal representation learning. Such a construction of the layered multi-label graph in Fig. 2(a) could bring three main favorable properties. First, the label-label connectivity allows direct and efficient high-order label interactions by simply adjusting the number of convolution layers in GCN. In addition, common nodes as attributes of the label nodes enable to encode graph community information in learned label representations, as nodes with identical labels tend to form a cluster or community. Lastly, the learned node representations can naturally preserve labels, label correlations and graph community information by taking label nodes as attributes of the node-node graph.

Fig. 2(b) shows the proposed MuLGCN model that contains two Siamese GCNs to simultaneously learn the label and node representations from the given multi-label graph, where the input feature vectors of both label nodes and common nodes are regularly updated during the training. First, the top-layer GCN learns label representations from the label-label-node graph through supervised single-label classification. Let $\mathbf{H} \in \mathbb{R}^{m \times d_f}$ be the input feature matrix of all m label nodes, $\mathbf{B}$ be the $m \times m$ adjacency matrix recording the co-occurrence relations between label nodes, and $\mathbf{F}$ be the $(n+m) \times (n+m)$ adjacency matrix of the input label-label-node graph. The first convolutional layer aggregates information from the neighborhood label nodes and the associated common nodes (*e.g.*, label node L1 in Fig. 2(a)), where the new d_e-dimensional label node feature matrix $\mathbf{L}^{(1)} \in \mathbb{R}^{m \times d_e}$ is computed by:

$$\mathbf{L}^{(1)} = \rho(\tilde{\mathbf{F}}^* \mathbf{H}^* \mathbf{W}_0^l) \tag{3}$$

where ρ is an activation function, such as the *ReLU* represented by $\rho(x) = \max(0, x)$, $\mathbf{W}_0^l \in \mathbb{R}^{d_f \times d_e}$ is a weight matrix for the first label-label-node GCN layer and $\mathbf{H}^* = [\mathbf{H}; \mathbf{X}]^T$ is a vertically stacked $(m+n) \times d_f$ feature matrix. $\tilde{\mathbf{F}}^*$ is a truncated normalized symmetric adjacency matrix obtained by:

$$\mathbf{F}^* = \mathbf{F} + \mathbf{I}_{m+n}; \tilde{\mathbf{F}} = \mathbf{D}_f^{-\frac{1}{2}} \mathbf{F}^* \mathbf{D}_f^{-\frac{1}{2}}; \tilde{\mathbf{F}}^* = \tilde{\mathbf{F}}[:m] \tag{4}$$

where $\mathbf{I}_{m+n}$ is the identity matrix, $\mathbf{D}_f$ is the degree matrix with $\mathbf{D}_{f,ii} = \sum_j \mathbf{F}^*_{ij}$. One layer GCN only incorporates immediate label node neighbors. When higher order label correlations need to be preserved, we can easily stack multiple GCN layers (*e.g.*, the layer number $k \geq 2$) by:

$$\mathbf{L}^{(k)} = \rho(\tilde{\mathbf{B}} \mathbf{L}^{(k-1)} \mathbf{W}_k^l) \tag{5}$$

where $\tilde{\mathbf{B}} = \mathbf{D}_b^{-\frac{1}{2}} (\mathbf{B} + \mathbf{I}_m) \mathbf{D}_b^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and $\mathbf{D}_{b,ii} = \sum_j (\mathbf{B} + \mathbf{I}_m)_{ij}$. The last layer output label embeddings (*e.g.*, assume we consider a two-layer GCN) have the same size as the total number of labels (e.g., $d_e = m$) and are through a *softmax* classifier to perform the single-label classification (e.g., each label node corresponds to the label itself in the classification) by:

$$\mathbf{O}^l = \tilde{\mathbf{C}} ReLU(\tilde{\mathbf{F}}^* \mathbf{H}^* \mathbf{W}_0^l) \mathbf{W}_1^l \tag{6}$$

$$\mathbf{Z}^l = softmax(\mathbf{O}^l) = \frac{\exp(\mathbf{o}^l)}{\sum_i \exp(\mathbf{o}_i^l)} \tag{7}$$

where $\mathbf{W}_0^l \in \mathbb{R}^{d_f \times d_h}$ and $\mathbf{W}_1^l \in \mathbb{R}^{d_h \times d_e}$ are the weight matrices for the first and second label-label-node GCN layers, respectively. We can conclude that the label representations are learned in a supervised manner, where the resulting label representations $\mathbf{O}^l \in \mathbb{R}^{m \times d_e}$ are in return used to predict the respective labels themselves. Let $\mathbf{Y}^l \in \mathbb{R}^{m \times m}$ be the one-hot

label indicator matrix of all m label nodes, the classification loss can be defined as the cross-entropy error computed by:

$$\mathcal{L}_1 = -\sum_{d=1}^{m} \mathbf{Y}_d^l \ln \mathbf{Z}_d^l \tag{8}$$

Then, the bottom-layer GCN learns node representations from the node-node-label graph. Similarly, in the first layer each convolution node aggregates information from the neighborhood common nodes and the associated attributed label nodes (*e.g.*, take node V2 in Fig. 2(a) as an example). Let $\mathbf{E}$ be the $(n+m)\times(n+m)$ adjacency matrix of the input node-node-label graph, the d_e-dimensional node embeddings output by the first GCN layer are computed as:

$$\mathbf{N}^{(1)} = \rho(\tilde{\mathbf{E}}^* \mathbf{X}^* \mathbf{W}_0^v) \tag{9}$$

where $\mathbf{W}_0^v \in \mathbb{R}^{d_f \times d_e}$ is a weight matrix for the first node-node-label GCN layer, $\mathbf{X}^* = [\mathbf{X}; \mathbf{Y}]^T$ and $\tilde{\mathbf{E}}^*$ is a truncated normalized symmetric adjacency matrix obtained by:

$$\mathbf{E}^* = \mathbf{E} + \mathbf{I}_{m+n}; \mathbf{E} = \mathbf{D}_e^{-\frac{1}{2}} \mathbf{E}^* \mathbf{D}_e^{-\frac{1}{2}}; \tilde{\mathbf{E}}^* = \tilde{\mathbf{E}}[:n] \tag{10}$$

where $\mathbf{D}_e$ is the degree matrix with $\mathbf{D}_{e,ii} = \sum_j \mathbf{E}_{ij}^*$. As in the label-label-node graph, we can also incorporate k-hop neighborhood information by stacking multiple GCN layers:

$$\mathbf{N}^{(k)} = \rho(\tilde{\mathbf{A}} \mathbf{N}^{(k-1)} \mathbf{W}_k^v) \tag{11}$$

where $\tilde{\mathbf{A}} = \mathbf{D}_a^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I}_n)\mathbf{D}_a^{-\frac{1}{2}}$ and $\mathbf{D}_{a,ii} = \sum_j (\mathbf{A}+\mathbf{I}_n)_{ij}$. The node embeddings $\mathbf{O}^v \in \mathbb{R}^{n \times d_e}$ output by the last layer have size m (e.g., $d_e = m$) and are passed through a sigmoid transformation to perform supervised multi-label classification with the collective cross-entropy loss (*e.g.*, the two-layer GCN are used in this paper) over all labeled nodes computed by:

$$\mathbf{O}^v = \tilde{\mathbf{A}} ReLU(\tilde{\mathbf{E}}^* \mathbf{X}^* \mathbf{W}_0^v) \mathbf{W}_1^v \tag{12}$$

$$\mathcal{L}_2 = -\sum_{i \in \mathbf{y}} \mathbf{Z}^v \tag{13}$$

where $\mathbf{W}_0^v \in \mathbb{R}^{d_f \times d_h}$ and $\mathbf{W}_1^v \in \mathbb{R}^{d_h \times d_e}$ are the weight matrices for the first and second node-node-label GCN layers, respectively, $\mathbf{y}$ is the set of node indices that have labels for supervised training. Let $\mathbf{Y}^v \in \mathbb{R}^{n \times m}$ be the one-hot label indicator matrix of all n common nodes, then $\mathbf{Z}^v$ is calculated as:

$$\begin{aligned}
\mathbf{Z}^v &= \mathbf{Y}_i^v \log(\sigma(\mathbf{O}_i^v)) + (1 - \mathbf{Y}_i^v) \log(1 - \sigma(\mathbf{O}_i^v)) \\
&= \mathbf{Y}_i^v \log\left(\frac{1}{1+\exp(-\mathbf{O}_i^v)}\right) + \left(1 - \mathbf{Y}_i^v\right) \log\left(\frac{\exp(-\mathbf{O}_i^v)}{1+\exp(-\mathbf{O}_i^v)}\right) \\
&= -\mathbf{Y}_i^v \log\left(1 + \exp(-\mathbf{O}_i^v)\right) \\
&\quad - \left(1 - \mathbf{Y}_i^v\right) \log\left(\mathbf{O}_i^v + \log\left(1 + \exp(-\mathbf{O}_i^v)\right)\right) \\
&= -\left(1 - \mathbf{Y}_i^v\right) \mathbf{O}_i^v - \log(1 + \exp(-\mathbf{O}_i^v)).
\end{aligned} \tag{14}$$

The above two aspects of representation learning for labels and nodes are trained together and impact one another by sharing the common classification labeling space $\mathbf{c}$ from the target graph G, and in the meantime a subset of input features, *i.e.*, through the attributed label nodes in the bottom-level node-node graph and the attributed common nodes in the top-level label-label graph. Let the total number of training epochs for MuLGCN be I, after N-epoch training of common node representations, the input feature matrix for the label-label-node graph will be updated:

$$\mathbf{X}_{new} = \rho(\mathbf{O}^v \mathbf{W}^v); \mathbf{H}^* = [\mathbf{H}; \mathbf{X}_{new}]^T, \tag{15}$$

Algorithm 1: Training MuLGCN

Input : A multi-label graph $G = (\mathbf{v}, \mathbf{e}, \mathbf{c}, \mathbf{A}, \mathbf{U}, \mathbf{X})$
Output: The node representations $\mathbf{O}^n = \{h_i\}_{i=1,\cdots,n}$
Initialization: $i = 0$, the training epoch I, the information updating frequencies M and N

```
while i ≤ I do
    Feed the label-label-node graph to train label representations;
    Feed the node-node-label graph to train node representations;
    if i%M = 0 then
        Update the feature matrix by Eq. (15);
    end
    if i%N = 0 then
        Update the feature matrix by Eq. (16);
    end
    Optimize L1 and L2 by the collective classification objective of Eq. (17);
    i = i +1.
end
```

In the meantime, after M-epoch training of label representations, the input feature matrix for the node-node-label graph will be updated:

$$\mathbf{H}_{new} = \rho(\mathbf{O}^l \mathbf{W}^l); \mathbf{X}^* = [\mathbf{X}; \mathbf{H}_{new}]^T \tag{16}$$

where $\mathbf{W}^v \in \mathbb{R}^{d_e \times d_f}$ and $\mathbf{W}^l \in \mathbb{R}^{d_e \times d_f}$ are weight matrices that are trained through the label-label-node graph learning in Eq. (6) and the node-node-label graph learning in Eq. (12), respectively. The collective training procedure for the MuLGCN model has been summarized in Algorithm 1.

4.2 Algorithm Optimization and Complexity Analysis

The node representation and label representation learning in Algorithm 1 are not independent, but depend on each other through shared embedding features learned from two reciprocally enhanced GCNs. In addition, the two level GCNs conduct two supervised classification tasks in the same labeling space: the top label-label-node GCN is doing a single-label classification and the bottom node-node-label GCN is doing a multi-label node classification. Finally, the global learning objective is to minimize the following collective classification loss:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{17}$$

In this paper, all weight parameters are optimized using gradient descent as in [15] and [31].

The training of MuLGCN is efficient in terms of the computational complexity. In this paper, we adopt a two-layer GCN and one-layer GCN for learning the node and feature representations, respectively. Since multiplication of the adjacency matrix (e.g., $\mathbf{A}$ for the node-node graph and $\mathbf{F}$ for the label-label graph) and feature matrix (e.g., $\mathbf{H}^*$ and $\mathbf{X}^*$ in Eqs. (6) and (12), respectively) can be implemented as a product of a sparse matrix with a dense matrix, the algorithm complexity of MuLGCN can be represented as $O((Ed_f d_h m + nd_f) + (Ld_f m + md_f))$, where n and m are the number of nodes and labels, E and L are the number of

TABLE 2: Dataset characteristics.

Items	BlogCatalog	Flickr	YouTube	MIR
# Nodes	10, 312	8, 052	22, 693	5, 892
# Edges	333, 983	4, 332, 620	192, 722	380, 808
# Labels	39	194	47	152
# Co-occur.	615	3, 716	1, 079	5, 074

edges in node-node-label graph and label-label-node graph, respectively. d_f is the dimension (for both nodes and labels) of input feature vectors. d_h is dimension of the hidden feature vectors produced in the first node-node-label GCN layer of all common nodes.

For most networks m, L and n are generally far more less than E (*e.g.*, as we will see in section V, for the Filckr dataset, E is 4,332,620, compared with m, L and n are merely 194, 3,716 and 8,052, respectively), therefore the complexity of MuLGCN is approximately equivalent to $O(Ed_f d_h m)$, which is the same as GCN. Meanwhile, since Eqs. (15) and (16) are not computed at each epoch (*e.g.*, every 50 epoch), the complexity for our model is still $O(Ed_f d_h m)$, the same theoretical asymptotic complexity as the GCN.

5 Experiments & Results

In this section, we compare the proposed approach against a set of strong baselines on four real-world datasets by conducting supervised node classification.

5.1 Benchmark Datasets

We collect three multi-label document networks [13], [23], BlogCatalog, Flickr, and YouTube, as the benchmark. We also use a multi-label image network called MIR[1]. They are described as follows.

BlogCatalog is a network of social relationships among 10,312 blogger authors (nodes), where the node labels represent bloggers' interests such as *Education*, *Food* and *Health*. There are 39 unique labels in total and each node may be associated with one or multiple labels. It is easy to find that users' labels of interest often interact and correlate with each other to enhance the affinities between blogger authors. For example, *food* is highly related with *Health* in real life, where two users have both labels *food* and *life* should be much closer compared with those whom only share either label *food* or label *life*. There are 615 co-occurrence (abbreviated as co-occur.) relationships (*e.g.*, correlations) among all 39 labels in this dataset.

Flickr is a photo-sharing network between users, where node labels represent user interests, such as *Landscapes* and *Travel*. There are 8,052 users and 4,332,620 interactions (*e.g.*, edges) among them. Each user could have one or multiple labels of interest from the same labeling space of 194 labels in total.

YouTube is a social network formed by video-sharing behaviors, where labels represent the interest groups of users who enjoy common video genres such as *anime* and *wrestling*. There are 22,693 users and 192,722 links between them. Each pair of linked users may share multiple identical labels out of the total 47 labels. The number of correlations between these labels is 1,079.

MIR is built by forming links between images sharing common metadata from Flickr. Edges are formed between images from the same location, submitted to the same gallery, group, or set, images sharing common tags, images taken by friends, etc. There are 5,892 nodes and each node represents a 500×375 RGB image that corresponds to one or more of the 134 classes.

The detailed statistic information of the above four multi-label networks is summarized in Table 2.

5.2 Comparative Methods

We compare the performance of the proposed method with the following state-of-the-art methods for multi-label node classification:

- **DeepWalk** [23] is a shallow network embedding model that only preserves the topology structures. It captures the node neighborhood relationships through random walks over the network and then derives node representations based on the SkipGram model.
- **LINE** [32] is also a structure preserving method. It optimizes a carefully designed objective function that preserves both the local and global network structures, compared with the DeepWalk that encodes only the local structures.
- **Node2vec** [25] adopts a more flexible neighborhood sampling process than DeepWalk to capture the node relationships. The biased random walk of Node2vec can capture second-order and high-order node proximity for representation learning.
- **GENE** [13] is a network embedding method that simultaneously preserves the topology structures and label information. Different from the proposed approach in this paper, GENE simply models labels as plain attributes to enhance structure-based representation learning process, whereas our model considers multi-label correlation and network structure for representation learning.
- **GCN** [15] is a state-of-the-art method that can naturally learn node relations from network structures and features, where each node forms its representation by adopting a spectral-based convolutional filter to recursively aggregate features from all its neighborhood nodes.
- **Text GCN** [31] is built on GCN that aims to embed heterogeneous information network. In this paper, we construct a heterogeneous node-label graph, where common nodes and label nodes are directly connected by their labeling relationships.
- **ML-GCN** [16] is specifically proposed for multi-label image classification. We adapt and adopt a similar way by constructing a directed label-label graph based on label co-occurrence relationships. The label representations learned by GCN on label-label graph are then incorporated into node representations by a dot-production operation for supervised node classification.
- **MuLGCN**$_{node}$ is a variation of the proposed MuLGCN model that removes attributes of common nodes from the label-label-node graph. Therefore, the community information is not preserved in this method.

1. https://snap.stanford.edu/data/web-flickr.html

- **MuLGCN**$_{1n}$ is a variation of the proposed MuLGCN model. The only difference with MuLGCN is that MuLGCN$_{1n}$ takes only one convolutional layer while learning from the node-node-label graph.
- **MuLGCN**$_{2l}$ is a variation of the proposed MuLGCN model, which adopts two consecutive convolution layers to learn from the label-label-node graph, compared with one layer in MuLGCN.
- **MuLGCN** is our proposed multi-label learning approach in this paper. It considers a two-layer graph structure–a top-level label-label-node graph which allows the preservation of label correlations and meanwhile a bottom-level node-node-label graph that enables the label correlation-enhanced node representation learning.

The above baselines can be roughly separated into three categories based on the types of information (*e.g.*, network structures and labels) and how it is incorporated in the graph embedding models. The first class belongs to methods that only preserve graph structures, including DeepWalk, Node2vec, LINE, GCN (*e.g.*, we use the structure-based identity matrix as the original features of all nodes). The second class includes GENE and Text GCN that preserve both the graph structures and label information, where the labels are modeled as plain attribute information to enhance structure-based representation learning. The proposed method MuLGCN and its variants (MuLGCN$_{node}$, MuLGCN$_{1n}$, and MuLGCN$_{2l}$) represent the third class, which not only preserve structural and label information, but also the correlations between labels.

It is worth noting that we designed three variations of MuLGCN (including MuLGCN$_{node}$, MuLGCN$_{1n}$, and MuLGCN$_{2l}$) to validate its performance under different settings. This allows us to fully observe MuLGCN's performance and conclude which part is playing major roles for multiple-label GCN learning. For examples, MuLGCN$_{node}$ examines whether common node representations can be used to enhance the label correlations modelling and representations learning. MuLGCN$_{1n}$ and MuLGCN$_{2l}$ examine the influence of the number of graph convolution layers for learning node-node-label and label-label-node graphs, respectively. In addition, when the label-label-node graph is not modeled, MuLGCN degrades to the basic GCN model that learns simply from the original graph between common nodes.

5.3 Experimental Settings

There are many hyper parameters involved. Some are empirically set [31] while others are selected through sensitivity study experiments. For MuLGCN, we use two-layer and one-layer GCNs to respectively learn from the node-node-label graph and the label-label-node graph. We test hidden embedding size, d_h, between 50 to 500, training ratios, α, of supervised labeled instances between 0.025 and 0.2, and updating frequencies N and M from 10 to 100, respectively. We also compare the performance of MuLGCN through differing numbers of GCN convolution layers (e.g., MuLGCN$_{1n}$ and MuLGCN$_{2l}$). For comparison, we set the learning rate η for gradient decent as 0.02, training epoch as 300, dropout probability as 0.5, and the default values of d_h, α, N and M as 400, 0.2, 50 and 50, respectively. For each image in the MIR dataset, we extract a CNN feature descriptor and the feature dimension for MIR is 152. The L_2 norm regularization weight decays are set as 0 and 0.005 for document network and image network, respectively. After selecting the labeled training instances, the rest is split into two parts: 10% as validation set and 90% for testing set.

It is necessary to mention that all baselines are set to conduct the multi-label node classification (e.g., each node can belong to multiple labels) under the same environmental settings. As metrics used in [23] and [32], we adopt Micro-F1 and Macro-F1 to evaluate the node classification performance, which are defined as follows:

$$\text{Micro} - \text{F1} = \frac{\sum_{i=1}^{\mathbf{c}} 2TP^i}{\sum_{i=1}^{\mathbf{c}} (2TP^i + FP^i + FN^i)} \tag{18}$$

$$\text{Macro} - \text{F1} = \frac{1}{|\mathbf{c}|} \sum_{i=1}^{\mathbf{c}} \frac{2TP^i}{(2TP^i + FP^i + FN^i)} \tag{19}$$

where $\mathbf{c}$ is the set of labels from the target graph G. TP^i, FN^i and FP^i denote the number of true positives, false negatives and false positives w.r.t the ith label category, respectively. All experiments are repeated 10 times with the average results and their standard deviations reported.

5.4 Experimental Results

Table 3 presents the comparative results of all methods with respect to the multi-label classification performance under the same environment settings, where the top three best results have been highlighted. From the table, we have the following main observations.

5.4.1 Shallow Network vs. GCN

Among all methods that encode only the graph topology structures, the shallow neural networks-based methods (e.g., DeepWalk, LINE and Node2vec) perform poorly with a wide gap compared with deep model GCN over all document networks, *i.e.*, on BlogCatalog network, the classification performance of GCN improved 30.6% and 70.9% over Node2vec w.r.t Micro-F1 and Macro-F1, respectively. Although LINE slightly performs better than GCN w.r.t Micro-F1, GCN improved by a large margin w.r.t Macro-F1 performance. This is because shallow models have limitations in learning complex relational patterns among nodes [1]. For example, although Node2vec relies on a carefully designed random walk process to capture the node neighborhood relationships, it cannot differentiate the affinities between a node and others within the same walk sequence. In comparison, GCN uses a more efficient way to constrain the neighborhood relations between nodes, where each node only interact with its neighbors in each convolution layer. Such a learning paradigm is more accurate to maintain the actual node relevance reflected by the edge links without introducing noisy neighborhood relationships as in Node2vec.

5.4.2 Label Correlation & Utilization

In terms of the performance of methods (GENE and Text GCN) that have incorporated the labels to enhance the structures modeling, we can find that GENE is built on DeepWalk to additionally preserve the label information.

TABLE 3: Multi-label classification performance comparison. The 1st, 2nd, and 3rd best results are bold-faced, italic-formatted and underscored respectively.

Metrics	Micro-F1 (%)				Macro-F1 (%)			
Methods	BlogCatalog	Flickr	YouTube	MIR	BlogCatalog	Flickr	YouTube	MIR
DeepWalk	29.19 ± 0.28	25.75 ± 0.13	26.19 ± 0.18	50.92 ± 0.17	19.22 ± 0.63	12.31 ± 0.17	10.03 ± 0.23	30.47 ± 0.80
LINE	30.79 ± 0.84	30.13 ±0.39	27.68 ± 0.11	51.44 ± 0.37	17.89 ± 1.26	16.24 ± 0.53	10.90 ± 0.30	27.55 ± 0.26
Node2vec	33.35 ± 0.69	34.51 ± 0.44	26.75 ± 0.25	34.95 ± 1.69	21.38 ± 1.76	19.72 ± 0.32	10.03 ± 0.29	15.07 ± 1.70
GENE	28.77 ± 0.02	29.44 ± 0.18	26.77 ± 0.22	51.80 ± 0.68	16.00 ± 1.06	14.35 ± 0.82	10.48 ± 0.72	30.90 ± 0.62
GCN	43.57 ± 0.11	40.36 ± 0.04	*44.44 ± 0.02*	51.09 ± 0.31	36.55 ± 0.22	24.77 ± 0.09	33.54 ± 0.08	33.27 ± 0.35
Text GCN	40.31 ± 0.49	*41.82 ± 0.33*	39.82 ± 0.53	52.23 ± 0.47	32.03 ±0.49	22.98 ± 0.40	*39.07 ± 0.11*	33.40 ± 0.40
ML-GCN	37.88 ± 0.48	36.06 ± 0.38	42.03 ± 0.62	51.57 ± 0.57	31.10 ±0.65	24.23 ± 0.78	28.25 ± 0.76	30.40 ± 0.34
MuLGCN$_{node}$	43.72 ± 0.31	39.99 ± 0.07	44.14 ± 0.05	52.52 ± 0.30	38.39 ± 0.38	22.70 ± 0.03	33.96 ± 0.18	35.06 ± 0.35
MuLGCN$_{1n}$	32.96 ± 0.01	31.83 ± 0.03	31.82 ± 0.07	**53.59 ± 0.28**	17.92 ± 0.02	16.89 ± 0.21	27.48 ± 0.33	*36.12 ± 0.41*
MuLGCN$_{2l}$	*43.86 ± 0.07*	38.41 ± 0.23	42.33 ± 0.03	*53.42 ± 0.33*	*38.63 ± 0.24*	*26.28 ± 0.30*	32.62 ± 0.11	**37.61 ± 0.51**
MuLGCN	**45.17 ± 0.20**	**43.74 ± 0.20**	**45.71 ± 0.02**	52.61 ± 0.10	**42.53 ± 0.72**	**30.71 ± 0.12**	**42.77 ± 0.31**	35.08 ± 0.44

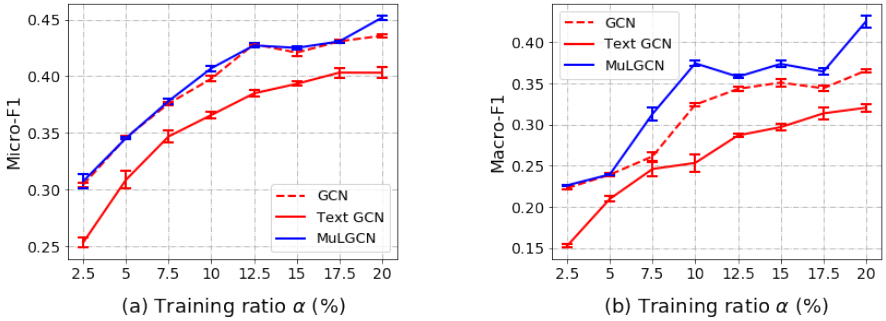


Fig. 3: Algorithm performance comparisons with respect to different percentage of training sample ratios (the $x-$axis denotes the ratio of training samples comparing to the whole network).

We can observe that GENE performs slightly better than DeepWalk on both Flickr and YouTube datasets. The reason is that each label is considered as the high-level summation of a group of similar nodes, thus can be used to supervise and distinguish the neighborhood affinities between nodes within the same random walk sequence. Nevertheless, LINE and Node2vec can perform better than GENE in most cases over three document networks, *i.e.*, the Micro-F1 performance of LINE and Node2vec on BlogCatalog increased 2.0% and 4.6%, respectively. The reason is probably that they adopted more efficient random walk process to capture node neighborhood relationships. In addition, as we can see from Table 3, the label-preserved model, Text GCN, has no advantages compared with the basic GCN model. The reason is probably that the labels are considered as attributes that have not been leveraged in a meaningful manner. In other words, only the labeled nodes have the attribute of labels in the supervised node representation leaning and training, the scattered labels could have become the noisy information to confuse the neighborhood relationships modeling between nodes.

We can observe that MuLGCN consistently outperforms the Text GCN in leaning multi-label graphs over all four datasets, although Text GCN has modeled the label correlations. The reasons are mainly threefold. First, MuLGCN allows immediate and efficient label correlation modeling without depending on common nodes, *i.e.*, labels directly interact with each other over the label-label graph in the proposed model. Second, it is common to use different numbers of convolutional layer to learn from the label and node graphs respectively, since the node-node graph is much more complicated than the label-label graph that involves simple label interaction patterns. To obtain a model that best fits the given label and node graphs of different scales, one can easily change the number of layers used by the two-layer graph modelings in MuLGCN independently. But this is hard for Text GCN to coordinate the layer settings that are most suitable to model node relations and label relations simultaneously in the node-label graph. Finally, in the structure design of MuLGCN, each label has preserved the community information by taking all related common nodes as attributes, which make the node relations modeling and the label correlations modeling more dependent on each other to optimize the global network representation learning, *i.e.*, the node representations of one label could refine the node representation learning of another correlated label, as we will demonstrate in the case study later.

5.4.3 MuLGCN vs. ML-GCN

Despite the fact that ML-GCN [16] builds a label-label graph to model label correlations similar to MuLGCN, it does not show any advantage compared with our MuLGCN model and even the basic GCN model for graph structure-based representation learning. There are two major reasons for this phenomenon. First, ML-GCN adopts a rigid label correlation matrix construction process, thereby it is difficult to decide the optimal hyper parameters in the matrix binarization

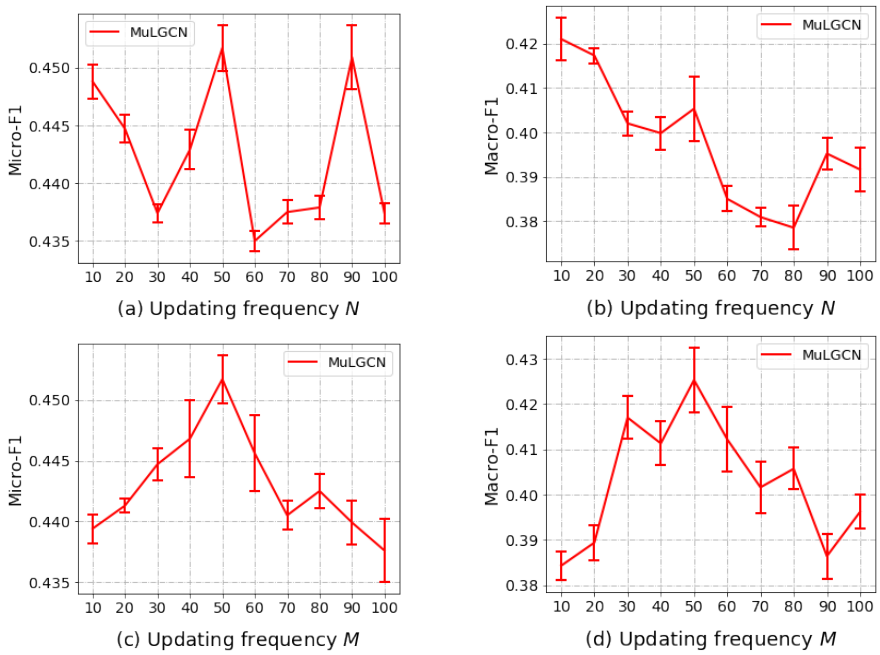


Fig. 4: Impact of the information updating frequencies.

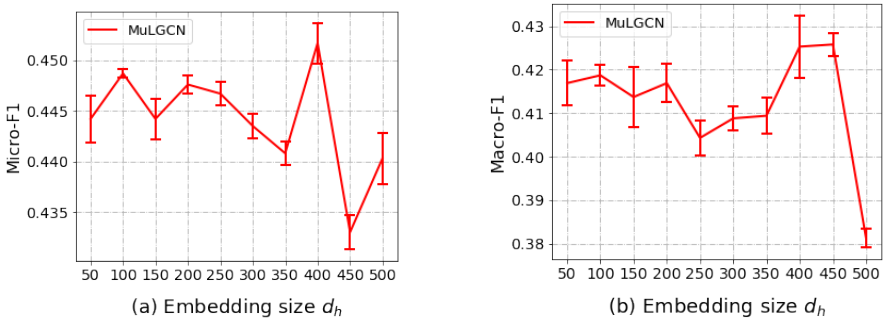


Fig. 5: Impact of the hidden node embedding size.

and smoothing steps [16]. Second, ML-GCN simply adopts a dot-production mechanism to combine label and node representations for unified node representation generation and classification at the end of the design, which is probably not efficient for multi-label learning of graph structured data. In comparison, MuLGCN uses a more dedicated design that allows labels and nodes to absorb helpful information from each other during learning through the two-layer label-label-node and node-node-label graph. The comparative results in Table 3 have demonstrated the superiority of our approach.

5.4.4 MuLGCN Variant Performance

In terms of the performance of different variations of the proposed MuLGCN, as shown in Table 3, we can conclude that MuLGCN is superior to MuLGCN$_{node}$, MuLGCN$_{1n}$ and MuLGCN$_{2l}$ on all three document networks, where the possible reasons are given as follows: 1) The comparison between MuLGCN and MuLGCN$_{node}$ demonstrates that taking the common nodes as attributes of the label-label network is beneficial, where labels could learn more enriched representations with encoding label correlations and the node communities to refine the neighborhood feature aggregation for node representation learning in the bottom-level node-node graph; 2) MuLGCN performs better than MuLGCN$_{2l}$, which illustrates a single-layer GCN is appropriate to model the label correlations, since compared with the node-node interactions, the label-label interactions are generally simple and explicit; 3) MuLGCN$_{2l}$ is inferior to MuLGCN, which demonstrates that exploring the high-order neighborhood relationships between nodes is important.

In Table 3, the results from the MIR network show that MuLGCN$_{1n}$ has the best Micro-F1 score, and also outperforms MuLGCN in Macro-F1 score. This may be because that the MIR network contains some irrelevant links between image nodes due to the coarse network construction process, *i.e.*, images from the same location have links, where the second-order neighborhood relationships preserved in MuLGCN could bring noise to some degree. In addition, we can observe from Table 3 that the ablation method MuLGCN$_{2l}$ has the best Macro-F1 score. There are two possible reasons. First, compared with the document networks, the image network MIR presents more frequent

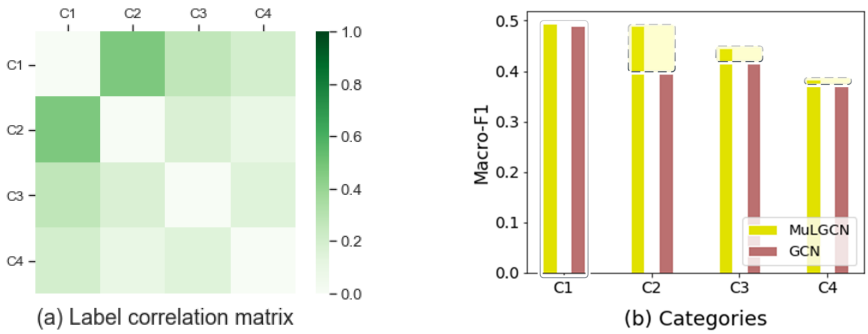


Fig. 6: (a) Pair-wise label correlation matrix. A higher gray intensity value (excluding main diagonal values) indicates a stronger correlation between two labels. (b) Classification performance (Macro-F1) with respect to different label categories.

label correlations, thereby it is useful to adopt the two-layer GCN to capture the complex correlation relations between labels. Second, each image from MIR dataset usually contains multiple objects reveled by the respective multiple labels. Therefore, compared with the document network, the global semantic of a multi-label image relies more on the interaction between labels, and a two-layer GCN is better than one-layer GCN to fully model the label interactions.

5.5 Parameter Sensitivity Study

We designed extensive experiments to test the sensitivities of various parameters between a wide range of values, such as the training ratio α of labeled nodes, the feature updating frequencies N and M, and the embedding size of the first hidden convolution layer while modeling the node-node-label graph. Fig. 3 shows the impacts of different portions of labeled training instances. In general, for all test models, we can observe that both the Micro-F1 (*e.g.*, Fig. 3(a)) and Macro-F1 (*e.g.*, Fig. 3(b)) performances increase with more labeled training nodes. This is reasonable since all these models adopt a supervised node representation learning and training manner, where the model parameters can be fully trained with larger labeled data [31]. Fig. 4 shows the influence of the input feature-updating frequencies controlled by N and M (*e.g.*, used in Eqs. (15) and (16)). We can see from Fig. 4(a) where the performance changes with N but no clear patterns can be observed in Micro-F1, while Fig. 4(b) shows an deceasing trend with larger values of N in the Macro-F1 scores. In comparison, from Fig. 4(c) and Fig. 4(d) the accuracy first increases then decreases with larger values of M w.r.t. both Micro-F1 and Macro-F1 scores. We also test the impact of node embedding size generated by the first convolutional layer with the trend shown in Fig. 5(a). The accuracy fluctuates before peaking at 400 and 450 w.r.t. Micro-F1 and Macro-F1 results, followed by a sharp decline.

5.6 Case Study

To illustrate how label correlations affect the multi-label graph learning performance, we present the classification results through four related label categories shown in Fig. 6. Fig. 7 presents their correlation matrix, where darker colors imply higher correlation between two corresponding labels, *i.e.*, c_1 and c_2 are highly correlated. We can see in Fig. 6 that MuLGCN and GCN perform similarly with respect to the node classification of category c_1. However, MuLGCN demonstrates better accuracy than GCN in classifying nodes in classes c_2, c_3 and c_4. Interestingly, the amount of accuracy improvement (*e.g.*, $c_2 > c_3 > c_4$) is, in fact, related to the strength of correlation of each class to c_1. This phenomenon might be caused by the fact that the L1 has an impact on its correlated labels during training. This also verifies that label interaction is critical for multi-label graph learning, and our proposed MuLGCN model can effectively capture and utilize this property.

6 Discussion

Multi-label graph structured data learning is a challenging problem not only because of the difficulties to efficiently model label correlations and implicit semantic interactions between labels and nodes over the graph [13], but also because of the significant influence caused by the tail label distribution problem in the data [33].

To model label correlations in the learning, we built an undirected label-label graph by label co-occurrence relationships. Since only a small portion of nodes with labels are known for supervised training, the label correlation matrix is relatively sparse and thus rational to preserve meaningful relationships between labels. When it comes to large-scale multi-label data, the label correlation matrix tends to be dense as it is likely that each label is bound with many others, which may introduce insignificant label correlations or even noise information [34].

The recent work [16] proposes to build a directed label-label graph where correlations between labels are either weighted by their co-appearing frequencies or neglected if the respective frequencies below a predefined threshold. The directed matrix is a good way to avoid dense and noisy label correlations, but it could meanwhile suffer from the tail label distribution problem, where a significant number of labels occur infrequently and neglecting them could degrade the learning performance [34], as demonstrated by the comparative results between MuLGCN and ML-GCN in this paper.

7 Conclusion

In this paper, we formulated a new multi-label network representation learning problem, where each node of the network may have multiple labels. To simultaneously explore

label-label correlation and the network topology, we proposed a multi-label graph convolution network (MuLGCN) to build two Siamese GCNs, a node-node-label graph and a label-label-node graph, from the multi-label network, and simultaneously carried out learning of node representation and label representation from the two GCNs. Because the two GCNs are unified to achieve one optimization goal, the learning of node representation and label representation are mutually beneficial to each other for maximum performance gain. Experiments on four real-world datasets verified the effectiveness of MuLGCN in combining labels, label correlations, and graph structures to enhance node representation learning and classification.

REFERENCES

[1] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *IEEE Trans. on Big Data*, vol. 6, no. 1, pp. 3–28, 2020.

[2] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[3] L. Liao, X. He, H. Zhang, and T.-S. Chua, "Attributed social network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2257–2270, 2018.

[4] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[6] S.-J. Huang and Z.-H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

[7] W. Bi and J. T. Kwok, "Multilabel classification with label correlations and missing labels," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[8] V. Kumar, A. K. Pujari, V. Padmanabhan, S. K. Sahu, and V. R. Kagita, "Multi-label classification using hierarchical embedding," *Expert Systems with Applications*, vol. 91, pp. 263–269, 2018.

[9] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multilabel image classification with regional latent semantic dependencies," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2801–2813, 2018.

[10] S. Burkhardt and S. Kramer, "Online multi-label dependency topic models for text classification," *Machine Learning*, vol. 107, no. 5, pp. 859–886, 2018.

[11] M. Shi, J. Liu, D. Zhou, and Y. Tang, "A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data," *IEEE Transactions on Services Computing*, 2018.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[13] J. Chen, Q. Zhang, and X. Huang, "Incorporate group information to enhance network embedding," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1901–1904.

[14] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 731–739.

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

[16] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[17] W. Liu and I. W. Tsang, "Large margin metric learning for multi-label prediction," in *Twenty-Ninth AAAI Conf. on Artificial Intelligence*, 2015.

[18] L. Wang, S. Chen, and H. Zhou, "Boosting up segment-level video classification performance with label correlation and reweighting."

[19] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[20] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.

[21] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[22] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification-revisiting neural networks," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2014, pp. 437–452.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[26] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[27] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[28] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1895–1901, 2016.

[29] T. M. Le and H. W. Lauw, "Probabilistic latent document network embedding," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 270–279.

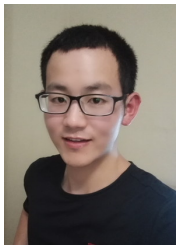
[30] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[31] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *arXiv preprint arXiv:1809.05679*, 2018.

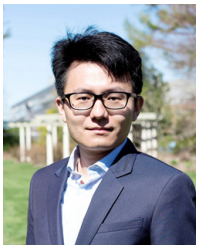
[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[33] C. Xu, D. Tao, and C. Xu, "Robust extreme multi-label learning," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1275–1284.

[34] T. Wei and Y.-F. Li, "Does tail label help for large-scale multi-label learning?" *IEEE Transactions on Neural Networks and Learning Systems*, 2019.



Min Shi (S'15) received his M.S. degree from the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. He is currently a Ph.D. candidate with the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, FL, USA. His research interests include data mining, machine learning, social networks, and service computing.



Yufei Tang (M'16) received the Ph.D. degree in Electrical Engineering from the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, in 2016. He is currently an Assistant Professor with the Department of Computer and Electrical Engineering & Computer Science, and a Faculty Fellow with the Institute for Sensing and Embedded Network Systems Engineering, Florida Atlantic University, Boca Raton, FL, USA. His research interests include machine learning, big data, smart grid, and cyber-physical systems.

Dr. Tang is an Early-Career Research Fellow of the National Academies Gulf Research Program (2019). He has received several awards including, the Steve Bouley and Rhonda Wilson Graduate Fellowship Award (2016), the Chinese Government Award for Outstanding Student Abroad (2016), the IEEE PESGM Graduate Student Poster Contest, Second Prize (2015), and the IEEE International Conference on Communications (ICC) Best Paper Award (2014).

Xingquan Zhu (SM'12) received the Ph.D. degree in computer science from Fudan University, Shanghai, China. He is a Full Professor with the Department of Computer and Electrical Engineering & Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests include data mining, machine learning, multimedia computing, and bioinformatics. Since 2000, he has authored or co-authored over 260 refereed journal and conference papers in these areas, including three Best Paper Awards and one Best Student Paper Award. Dr. Zhu is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2008 to 2012, and from 2014 to date. Since 2017, he has been an Associate Editor of the ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATA.

Jianxun Liu received the M.S. and Ph.D. degrees in Computer Science from the Central South University of Technology in 1997 and Shanghai Jiao Tong University in 2003, respectively. He is currently a Professor and the Dean of the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, China. His research interests include service computing, cloud computing, and big data. He has published more than 70 papers in peer-reviewed international journals and conferences, such as IEEE Transactions on Services Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Knowledge and Data Engineering, International Conference on Web Services, and International Conference on Services Computing.