Neurocomputing 532 (2023) 94–105

Contents lists available at ScienceDirect

Neurocomputing

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

Cluster-aware multiplex InfoMax for unsupervised graph representation learning

Xin Xu [a], Junping Du [a,*], Jie Song [b], Zhe Xue [a], Ang Li [a], Zeli Guan [a]

[a] Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] FreeWheel, Beijing 100026, China

ARTICLE INFO

Article history:
Received 23 November 2022
Revised 16 January 2023
Accepted 16 February 2023
Available online 21 February 2023
Communicated by Zidong Wang

Keywords:
Graph representation
Unsupervised learning
Contrastive learning
Mutual information maximization

ABSTRACT

Unsupervised graph learning aims to learn an encoder that embeds high-dimensional nodes into compact continuous vectors and preserves the topological and semantic features simultaneously without using any label information. Recently, contrastive learning (CL) on graph learning revives the traditional InfoMax principle and generates two views of the input graph randomly and then maximizes the agreements between them. However, the stochastic augmentation of a graph leads to two problems that need to be solved. Firstly, it ignores the role of some essential nodes and discriminating feature dimensions on the graph and may decrease the informativeness of the generated view by removing these crucial edges. Secondly, there are multi-level substructures of a graph that can be exploited and utilized for the network encoder's topological learning. This paper proposes Cluster-Aware Multiplex InfoMax (CAMI) for unsupervised graph representation learning. We apply an adaptive graph augmentation scheme on both topological and feature dimensions to generate graph views without damaging the vital graph structure. To encourage the network encoder to capture more underlying node interactions, we additionally increase a mutual information maximization constraint between the node's representation and multi-level graph summaries. Extensive experimental results on seven realistic datasets with different tasks prove the CAMI framework's effectiveness.

1. Introduction

Graph is quite common in the real-world because it can effectively abstract and model objects based on various connections. Since it is a modeling of reality, each node in the graph is independent of the other nodes, and the correlation between nodes contains rich information but is complex to analyze directly. On the other hand, the graph topology usually has no fixed patterns like images [1–3] or speech signals [4], which makes the traditional convolutional process difficult to operate on graphs. Much research has been done on graphs to observe this non-Euclidean characteristics and the expressive potential of graphs has recently been reconciled by Graph Neural Networks (GNN)-based [5–10] representation learning. Similar to how grid data is processed, variable-size permutation-invariant graphs are processed by GNNs to produce low-dimensional vectors that include details on the ascribed features as well as the graph's structure. The learned representations of nodes are commonly employed in downstream tasks including community discovery, anomaly detection, and recommendation system, among others.

However, the majority of effective GNN-based models are supervised and need a lot of labeled data for training, which is not practical in many situations because data labeling requires a lot of time and labor-intensive labors. Unsupervised graph learning [11,12] techniques are intended to learn node representations without labeling information while also exploring the features of the graph itself. Contrastive Learning (CL) has achieved great success by applying the mutual information maximization principle (InfoMax) [13] between the initial feature and its corresponding learned representation. Driven by the scalable estimation of mutual information through Mutual Information Neural Estimation (MINE) [14], and the representation learning of high-dimensional data introduced by Deep InfoMax (DIM) [15], the Deep Graph Infomax (DGI) [16] successfully combines the feature extraction ability of GNNs and the information constraint capacity of InfoMax principle to learn graph embeddings in an unsupervised manner. It generates another view of the original graph by simply shuffling the rows of the node feature matrix and applying a contrastive discriminator to distinguish whether a paired example is a

* Corresponding author.
E-mail address: junpingd@bupt.edu.cn (J. Du).

https://doi.org/10.1016/j.neucom.2023.02.036
0925-2312/

positive or negative pair. Every node is prompted to take note of the overall characteristics of the entire graph by the Mutual Information (MI) maximization between its representation and the global graph summary embedding. By comparing the input feature representation and the output representations of nodes without any clearly data augmentation, GMI [17], which follows DGI, introduces a more direct method to directly derive MI.

Despite its considerable success, contrastive learning for graphs still needs further investigation because it is limited by three issues: (1) The assumption of independent identical distribution of data (*i,i,d*) underlies the traditional method of augmentation of grid data (for example, photographs), which is invalid for data with a graph-structure. The most important characteristics of a graph are the topological structure and the attributed feature information. Simply shuffling the feature matrix results in deficient property corruption for generating diverse views of the graph. (2) Both a graph's adjacency matrix and the nodes' feature distribution are high-dimensional and frequently suffer from data sparsity. The edges that link important nodes and the feature dimensions that contain more information should be specially treated when performing data augmentation. Hence, the graph augmentation scheme should effectively guide the model to identify these edges and feature dimensions. (3) The MI maximization between a node's representation and the graph embedding ignores the fact that there are substructures in a graph that can be captured. A node tends to belong to a cluster that represents topologically nearby nodes as well as a cluster that represents semantically close nodes. The third difficulty in creating an efficient unsupervised graph learning model is how to capture the information content of these clusters and augment the structural information in a node representation.

In order to overcome the aforementioned difficulties, this study proposes Cluster-Aware Multiplex Infomax for unsupervised graph representation learning (CAMI). The proposed framework is made up of two main components: (1) *An adaptive graph augmentation scheme* that generates diverse graph views based on operations on both graph structure and feature dimension. With the adaptive graph augmentation scheme, unimportant edges are removed, and less informative feature dimensions are masked in the generated view, which encourages the network encoder to learn from the most representative graph structures and the inherent semantic features. (2) In each view, we use *a cluster-aware infomax learning* to concurrently optimizes the mutual information concerning the graph-level summary and cluster-level summaries. With the cluster-aware infomax learning, the learned node representation is enriched with more structural information and more robust to topological corruption, which is beneficial for the downstream tasks. In summary, the CAMI has the following contributions:

- We propose an unsupervised graph representation learning framework with adaptive graph augmentation. The model generates two correlated views concerning both the topological information and feature distribution, which improve the model's ability to resist interference in structural and feature aspects.
- We explore the InfoMax principle in each view and maximize the mutual information between node features with not only a graph-level summary but also a cluster-level summary. The multi-level MI maximization enriches the nodes' features with underlying substructure features.
- We perform unsupervised graph representation learning on seven different real-world datasets and verify the node representations on the node classification task, link prediction task, and node clustering task. All the experimental results demonstrate that the proposed CAMI can characterize the feature information, catch the underlying cluster-level summaries of the graph, and obtain effective node representations that work well on downstream tasks.

2. Related Works

2.1. Graph Representation Learning

After deep learning made significant strides, Graph Neural Networks (GNNs) has developed into a potent tool for graph analysis. Many graph representation learning algorithms based on GNNs have been presented, and their results are more encouraging than those of conventional random-walk-based techniques and factorization-based embedding methods [18–22]. Most GNNs follow the pipeline of AGGREGATE-COMBIN-PREDICTION, which first aggregates a node's neighborhood information, then combines these independent features with a summary function, and the last predictions are made on the aggregated information. The structure information and attributed features are collected and passed from multi-hop neighbors within the iterative aggregate and combined strategy. Various models have emerged with different aggregation designs and combined schemes to address diverse problems. For example, for semi-supervised graph data classification task, Graph Convolutional Networks (GCN) [7] employ a localized first-order approximation of spectral graph convolutions. Graph Attention Networks (GAT) [6] assign separated weights to each node in a neighborhood with the attention mechanism, which requires no costly matrix operations or upfront graph structure information. AdaGCN [23] incorporating AdaBoost into the GCN to extract knowledge from high-order neighbors with the shared base neural network architectures among all layers. Despite these models' variance in network structure, their superior performance largely depends on the supervised or semi-supervised training scheme, while sufficient node labels are often not accessible in the real world. Unsupervised graph learning techniques examine the properties of the graphs without the requirement for label information and provide a wider range of possible applications. The well-known GraphSAGE [24] presents an inductive framework for obtaining node representations in an unsupervised situation using a random-walk-based objective. GAE [25] introduces a graph auto-encoder framework for unsupervised graph-structure learning, which can be used for link prediction tasks. A critical challenge for unsupervised graph learning that how to take full advantage of the topological and attributed features of the graph is still under-explored.

2.2. Contrastive Learning

Due to its impressive performance in many tasks, contrastive learning has become an integral approach to unsupervised learning. It aims to learn an encoder to obtain discriminative representations by a "contrastive" loss which maximizes the similarities of positive pair examples and the distance of negative pair examples. This learning paradigm achieves promising performance in visual tasks [26–28] since images can be pre-processed with multi-stage data augmentation like rotation, color distortion, random flip, and cropping, which naturally generates positive and negative samples for training. However, the non-Euclidean property makes it challenging to generate negative samples for graph-structure data. Driven by the success of scalable neural estimator [14] for estimating mutual information through training a statistical network as a classifier to determine whether the sample is generated from joint distribution or their product of marginal distribution, the Deep Graph InfoMax (DGI) [16] marries the contrastive learning to deep graph representation learning. It generates negative samples by shuffling the node feature matrix and estimating the

mutual information between a node's representation and the graph summary representation. The remarkable performance of DGI has enlightened many works to explore the InfoMax principle [13] with contrastive learning. MVGRL [29] performs DGI-like objectives with the augmented views and introduces a discriminator that compares node representations from one view to graph representations from another, and vice versa. CommDGI [30] applies the mutual information maximization technique in graphs to capture neighborhood and community information in an unsupervised manner. InfomaxANE [31] encoded the attributed network's topological information and attributed feature using constrained objective to enforce nodes to learn different local embeddings from the perspective of mutual information. GMI [17] calculates the connection between original graph features and derived high-level representations using learnable weights for both graph proximity and feature space similarity. Unlike these works, we consider an adaptive graph augmentation scheme that keeps fundamental topological and semantic features in the generated views for contrastive learning. We also adapt the infomax principle in each view to capture the underlying structure of a graph.

3. Methodology

3.1. Notation and Problem Definition

Throughout this paper, we follow the commonly used notations and denote scalars as lowercase letters (e.g., v), vectors as bold lowercase letters (e.g., $\mathbf{x}$), matrix as bold uppercase letters (e.g., $\mathbf{X}$), sets as calligraphic fonts (e.g., $\mathcal{G}$).

We use the quadruple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}\}$ to represent a graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the node set and N indicates the number of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$represents the edge set. The adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$then can be used to represent the edge set and $\mathbf{A}_{ij} = 1$ indicates node v_i and v_j has an edge connected, otherwise $\mathbf{A}_{ij} = 0$. The feature matrix is expressed as $\mathbf{X} \in \mathbb{R}^{N\times D}$, where D is the dimension of the feature vector, and the i-th row the matrix is $\mathbf{x}_i \in \mathbf{R}^D$ that represents the node feature of node v_i.

The goal of graph representation learning is to train a network encoder $f_\theta : \mathbb{R}^{N\times d} \times \mathbb{R}^{N\times N} \rightarrow \mathbb{R}^{N\times d'}$ that can map each node $v \in \mathcal{V}$ into a low-dimensional continuous bedding space $\mathbf{h}_v \in \mathbb{R}^{d'}$, the node embedding's dimension d' is usually much smaller than the original dimension $d : d' \ll d$. All the learned vector $\mathbf{h}$ then form the matrix $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\}$ that indicates the learned high-level representations for all nodes. The learning process does not rely on any node label information, and the resulting representation $\mathbf{H}$ can then be used in downstream tasks.

We summarize the frequently used notations in Table 1.

Table 1
Description for frequently used notations.

Notation	Description
$\mathcal{G}$	Graph
$\mathcal{V}$	Node set
$\mathcal{E}$	Edge set
$\mathbf{A}$	Adjacency matrix
$\mathbf{X}$	Feature matrix
$\mathbf{H}$	Embedded node representation matrix
v_i	Node i in graph $\mathcal{G}$
d	Dimension of the node feature
d'	Dimension of the embedded node representation
$\mathbf{x}_i$	Node feature for node i
$\mathbf{h}_i$	Embedded node representation for node i
$\mathbf{s}$	The representation of graph summary
f_θ	Graph encoder
$\mathcal{D}$	Discriminator
$\mathbf{W}$	Parameters of the trainable matrix
$\mathbf{u}_k$	The representation of cluster summary
$\mathbf{z}_i$	The specific cluster summary representation for cluster i
c_n	A cluster
D	Degree matrix
I	Identity matrix
$I(X;Y)$	Mutual information between X and Y

3.2. Overall Framework

We propose a Cluster-Aware Multiplex Infomax framework for graph learning, inspired by recent breakthroughs in multi-view contrastive learning for visual representation learning and mutual information neural estimates. Our approach first performs adaptive graph augmentation that generates two correlated views. Then, we use cluster-aware mutual information estimation to optimize the mutual information between each node's representation and the representation of different levels of graph summary. No node label information is used in any learning phrase, and the self-supervised strategy ensures the multi-view generator and network encoder take full advantage of the graph. Fig. 1 displays the CAMI's entire architecture. Our model is made up of two primary components:

- An adaptive multi-view generating scheme that generates correlated views of the same graph. The data augmentation is performed on graph topology and node features, encouraging the model to emphasize informative nodes and feature dimensions and neglect noise information.
- A cluster-aware infomax learning in each view simultaneously maximizes the mutual information between each node's representation and the representation of graph-level summary and cluster-level summaries. This learning process enabled the node representations to capture vital information and nodal relationships even when the graph is corrupted.

Network Encoder. We apply a one-layer GCN as the network encoder to obtain the node representations. A GCN layer is formally defined as:

$$f(\mathbf{A}, \mathbf{X}) = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}\right) \tag{1}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix, $\hat{\mathbf{D}} = \sum_i \hat{\mathbf{A}}_i$ is the degree matrix, σ is the activation function like $ReLU(\cdot) = max(0, \cdot)$ or *sigmoid* function, and $\mathbf{W}$ is the weight matrix to be trained. The final node representation can be obtained as $\mathbf{H} = f_\theta(\mathbf{A}, \mathbf{X}) = \sigma\left(\sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}\right), \mathbf{A}\right)$. To capture the fundamental feature distribution in a graph, we employ a common network encoder across multiple views, as illustrated in Fig. 1.

Graph InfoMax. Two random variables X and Y's mutual information is defined as:

$$I(X;Y) = H(X) - H(X|Y) = \int_{XY} log\frac{dP_{XY}}{dP_X \times dP_Y} dP_{XY} \tag{2}$$

where $H(X)$ indicates the entropy of $X, H(X|Y)$ indicates the entropy bought by Y when X is given, P_{XY} represents the joint distribution of X and Y, and P_X and P_Y are the corresponding marginal distribution of X and Y.

Following DGI [16], the basic idea of training the network encoder with the InfoMax principle is to maximize the mutual information between nodes' patch representation $\mathbf{h}_i$ (local) and the representation of graph summary $\mathbf{s}$ (global). This training strategy encourages the encoder the aggregate more graph-level informa-

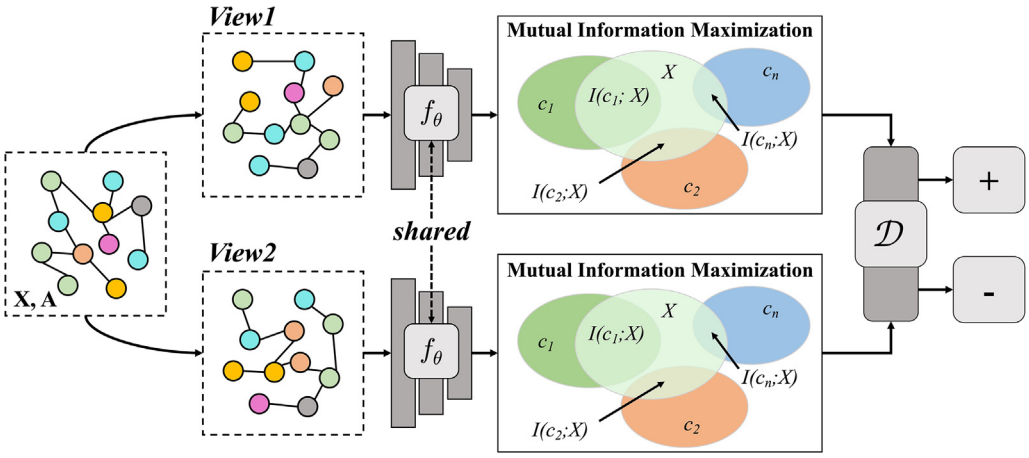

Fig. 1. Framework overview.

tion for a node, and the noise information in a node's vicinity should not be encoded. However, calculating the mutual information of continuous vectors precisely is difficult. The Jensen-Shannon MI estimator is frequently used to maximize MI's lower bound between the joint distribution and the marginal products. A discriminator is developed and employs a typical binary cross-entropy (BCE) loss across samples from the (positive) joint distribution and the (negative) product of marginals. The general objective is defined as:

$$\mathcal{L}_{graph} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{A},\mathbf{X})}[log\mathcal{D}(\mathbf{h}_i,\mathbf{s})] + \sum_{j=1}^{M}\mathbb{E}_{(\tilde{\mathbf{A}},\tilde{\mathbf{X}})}\left[log\left(1-\mathcal{D}\left(\tilde{\mathbf{h}}_j,\mathbf{s}\right)\right)\right]\right) \tag{3}$$

where $\mathbf{s}$ is the presentation of graph summary given by a *readout function*. $\tilde{\mathbf{h}}_j$ is the node representation obtained from an alternative graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}}, \tilde{\mathbf{A}}, \tilde{\mathbf{X}})$, and the shuffuled graph feature $(\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$ can be obtained with a *corruption function* that alters the topology and feature distribution of the original graph. $\mathcal{D}$ is the discriminator that provides the likelihood score assigned to this path-summary pair. By maximizing the mutual information between $\mathbf{h}_i$ and $\mathbf{s}$, the resulting patch representations are supposed to maintain sufficient mutual information with the global graph summary. The *graph infomax* encourages a node to aggregate the global graph-level information of the whole graph. However, there are also some hidden substructures underneath the whole graph that a node tends to belong to, such as clusters. Thus, we adapt the graph infomax to cluster-level representation and maximize the mutual information between a node and a cluster-level substructure summary to allow a node representation to capture richer topological information of a graph. On the other hand, the multi-level graph infomax is performed on the adaptively generated two related views of the graph, where irrelevant topological information and feature dimensions have been eliminated. By performing the cluster-aware infomax on multiple related views, we are obtaining node representations encoded with rich topological information and nodal interactions, which is beneficial to downstream tasks.

3.3. Adaptive Multi-View Generating

Unlike visual representation learning, which performs data augmentation on images by standard cropping, rotating, distorting colors, and so on, generating different views of the graph is still under-explored. The most important information of graph-structure data lies in the topology and the attributed feature. Thus, the augmentation function is supposed to preserve the crucial substructure and highlight the informative feature dimensions. We propose an adaptive multi-view generating scheme in the CAMI model that tends to retain the critical structures and ignore the unperceptive feature dimensions. We perform structure-wise augmentation by removing edges with a probability function that measures the importance and perform feature-wise augmentation by masking some of the dimensions with zeros. In conjunction with these two augmentation functions, the graph's generated view can capture the original graph's underlying topological and semantic schema.

Structure-wise augmentation. The graph structure is the topological link between nodes. We randomly remove edges in the graph for structure-wise augmentation to generate a correlated view. In the adaptive graph augmentation scheme, the generated view should contain important edges that reflect the graph's crucial structure, so the edge delete function should automatically recognize the importance of an edge. For an edge between node u and v, we assign it with a probability to represent the possibility it will be maintained in the generated view and denote it as $p_{e_{(u,v)}}$. The modified graph $\tilde{\mathcal{G}}$'s edge set $\tilde{\mathcal{E}}$ is a subset of the original whole graph.

An edge $e_{(u,v)}$ is a connection between two nodes, to qualify its importance, we use node centrality to measure it. Given the node centrality of a node $\varphi(v)$, the edge's centrality is defined as the mean of two connected nodes: $w_{e_{(u,v)}} = \frac{1}{2}(\varphi(u) + \varphi(v))$. Within a graph, some nodes have loads of neighborhood nodes while some nodes are at the network's periphery and have few connections, which varies the node centrality in a wide range. To mitigate the impact of the unequal distribution of node centrality, we set $s_{e_{(u,v)}} = logw_{e_{(u,v)}}$ and compute the edge importance with a normalization step:

$$p_{e_{(u,v)}} = \frac{s_{max} - s_{e_{(u,v)}}}{s_{max} - s_{average}} \tag{4}$$

where s_{max} is the maximum of $s_{e_{(u,v)}}$ and $s_{average}$ is the average of $s_{e_{(u,v)}}$.

In network analysis, the node centrality $\varphi(\cdot)$ has different measurements from different aspects. For example, *Node Degree* [32] is a basic indicator and reflects the number of neighborhoods in a network. It can be formalized as:

$$d_i = \sum_{j=1}^{N}\mathbf{A}_{ij} \tag{5}$$

where $\mathbf{A}_{ij}$ is an item in the adjacency matrix $\mathbf{A}$, if there is an edge between node i and node j, $\mathbf{A}_{ij} = 1$, otherwise, $\mathbf{A}_{ij} = 0$.

Another widely-used node centrality measurement is *PageRank Centrality* [32]. The PageRank algorithm was first introduced as Google's page ranking algorithm and has now been used to evaluate node importance in many works. The basic idea is that if a node is linked to many other nodes, it is more important, and the PageRank value will be relatively high. PageRank is calculated as:

$$PR_i = c\sum_{j\in N_i}\frac{PR_j}{d_j} \tag{6}$$

where N_i is the neighborhood the node i, d_j is the degree of node j, PR_j is the PageRank value of node j, and c is a constant.

We use a hyperparameter α in the view generating process to control the overall probability of cutting an edge. Then the probability of an edge been remained in a generated view is $1-\alpha p_{e_{(u,v)}}$.

Feature-wise augmentation Adding salt-and-pepper noise to an image is commonly used in image processing. Similarly, we add zeros to some feature dimensions to obtain the feature-wise augmentation. The feature distribution of the network is generally high-dimensional and suffers from data sparsity, so the adaptive feature-wise augmentation scheme should also reflect the informative dimensions and mask those less discriminative ones. The feature-wise augmentation is performed with a random vector $\tilde{\mathbf{m}} \in \{0,1\}^d$ that each dimension is sampled from a Bernoulli distribution independently. The generated feature matrix then can be calculated as:

$$\tilde{\mathbf{X}} = \{\mathbf{x}_1 \circ \tilde{\mathbf{m}}, \mathbf{x}_2 \circ \tilde{\mathbf{m}}, \ldots, \mathbf{x}_N \circ \tilde{\mathbf{m}}\} \tag{7}$$

where $\circ$ is the element-wise multiplication.

We presume that the feature dimensions that appear often with influential nodes are likewise significant. So, we use the node centrality $\varphi(\cdot)$ to calculate a feature dimension t's importance w_t:

$$w_t = \sum_{u\in\mathcal{V}}|\mathbf{x}_{\mathbf{u_t}}|\cdot\varphi(u) \tag{8}$$

where $\mathbf{x}_{\mathbf{u_t}}$ represents the t-th dimension of node u's feature vector and $\varphi(u)$ the node centrality of node u. Similar to the structure-wise augmentation, the feature importance is first set as $s_t^f = logw_t$ a normalization step is performed on all feature dimensions to obtain the probability score that indicates the significance of the feature:

$$p_t = \frac{s_{max}^f - s_t^f}{s_{max}^f - s_{average}^f} \tag{9}$$

where s_{max}^f is the maximum value of s_t^f and $s_{average}^f$ is the average value of s_t^f. A hyperparameter β is used to control the overall probability of the feature dimension being masked.

For the adaptive multi-view generating, we jointly use the structure-wise augmentation and the feature-wise augmentation with different probabilities of removing some edges and masking some feature dimensions to generate diverse graph views. With the node centrality calculation and adaptive probability measurement, important edges representing crucial topological structures are kept, and noisy feature dimensions that reduce the feature discriminativeness are wiped out in the generated graph.

3.4. Cluster-Aware Infomax Learning

For each generated view, we perform cluster-aware infomax learning by maximizing the mutual information between a node's representation and two levels of graph representation, i.e., a graph summary that contains the whole graph's information and cluster summaries that aggregate from different clusters. The MI maximization between the full graph enables the nodes to discover and preserve similarities across the graph, and the MI maximization between the substructure of the graph enables the nodes to embody various structural properties.

The mutual information estimation and maximization is processed with a discriminator function to distinguish a pair of representations $(\mathbf{h}_i, \mathbf{s})$ is sampled from the true graph $\mathcal{G}$ (positive) or from a fake input $\tilde{\mathcal{G}}$ (negative). The true graph is one of the generated views of the original graph, and the fake input is obtained by randomly shuffling the corresponding feature matrix's rows. Both true graph and disrupted graph are applied to the network encoder f_θ to obtain node representations. The global summary of a graph $\mathbf{s} \in \mathbb{R}^{1\times d'}$ is obtained by a *Readout function* that computes the average of all node representations:

$$\mathbf{s} = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_i\right) \tag{10}$$

where N is the number of nodes in the graph, and σ is the logistic sigmoid nonlinearity. The discriminator gives the scores of possibilities of $(\mathbf{h}_i, \mathbf{s})$ of serving as a positive example and is formally defined as:

$$\mathcal{D}(\mathbf{h}_i, \mathbf{s}) = \sigma\left(\mathbf{h}_i^\top \mathbf{W}\mathbf{s}\right) \tag{11}$$

where σ represents the logistic sigmoid nonlinearity and $\mathbf{W}$ represents the trainable scoring matrix. For a node representation $\tilde{\mathbf{h}}_i$ obtained from the fake graph, it forms the negative pair with the true graph's summary $(\tilde{\mathbf{h}}_i, \mathbf{s})$. The discriminator is trained to give higher scores to positive pair samples where a node representation is contained within the global summary. The objective for graph-level mutual information estimation and maximization is expressed as Eq. 3.

Although no label information is available during training, we believe the node representations in the embedding space naturally show a property that nodes belonging to the same class would gather around. In other words, clusters in the embedding space reflect the inherent topological structure. To cluster nodes into K different clusters, we employ the unsupervised K-means algorithm. As in ClusterNet [33], the cluster summary $\mathbf{u}_k$is obtained and implemented with a differentiable variant of the K-means clustering. Based on cluster modularity maximization, the cluster centers are updated using the ClusterNet layer in an end-to-end differentiable way, and each cluster center is defined with:

$$\mathbf{u}_k = \frac{\sum_i r_{ik}\mathbf{h}_i}{\sum_i r_{ik}} \quad k = 1, \ldots, K \tag{12}$$

and

$$r_{ik} = \frac{exp(-\gamma \mathrm{sim}(\mathbf{h}_i, \mathbf{u}_k))}{\sum_k exp(-\gamma \mathrm{sim}(\mathbf{h}_i, \mathbf{u}_k))} \quad k = 1, \ldots, K \tag{13}$$

where $\mathrm{sim}(\cdot,\cdot)$computes the degree of similarity between two occurrences and γ denotes the inverse-temperature hyperparameter. Then, we have K cluster summaries for substructure $\mathbf{u}_k \in \mathbb{R}^{1\times d'}$ with $k = 1, 2, \ldots, K$. Thus, we denote $\mathbf{z}_i$ as the cluster summary and compute it with weighted average summaries of the clusters to which node v_i belongs, as:

$$\mathbf{z}_i = \sigma\left(\sum_{k=1}^{K} r_{ik}\mathbf{u}_k\right) \tag{14}$$

where σ is a logistic sigmoid nonlinearity, r_{ik} is the degree that node n_i belongs to cluster $k, \sum_k r_{ik} = 1$ and $\mathbf{z}_i \in \mathbb{R}^{1\times d'}$.

To ensure a node transmits more structured information about a cluster, we maximize the mutual information between a node's

representation and the clusters' summary representation. For cluster-level mutual information estimation and maximization, we get positive pairings by pairing an actual graph node representation $\mathbf{h}_i$ with its matching cluster summary $\mathbf{z}_i$ and negative pairs come from the node representation $\tilde{\mathbf{h}}_i$ from the fake graph and the cluster summary $\mathbf{z}_i$. For simplicity and computational saving, an inner product similarity followed by a logistic sigmoid nonlinearity $\sigma(\cdot)$ is utilized as the discriminator $\mathcal{D}_{cluster}$:

$$\mathcal{D}_{cluster}(\mathbf{h}_i, \mathbf{z}_i) = \sigma\left(\mathbf{h}_i^{\top}\mathbf{z}_i\right) \tag{15}$$

Thus, the objective for cluster-level mutual information estimation and maximization is:

$$\mathcal{L}_{cluster} = \sum_{i=1}^{N}\mathbb{E}_{(\mathbf{A},\mathbf{X})}[log\mathcal{D}_{cluster}(\mathbf{h}_i, \mathbf{z}_i)] + \sum_{j=1}^{N}\mathbb{E}_{(\tilde{\mathbf{A}},\tilde{\mathbf{X}})}\left[log\left(1 - \mathcal{D}_{cluster}\left(\tilde{\mathbf{h}}_i, \mathbf{z}_i\right)\right)\right] \tag{16}$$

and the overall objective for one of the generated views is in conjunction with the graph-level objective and cluster-level objective:

$$\mathcal{L}_1 = \lambda\mathcal{L}_{graph} + (1-\lambda)\mathcal{L}_{cluster} \tag{17}$$

where λ is a parameter that determines each component's importance in a view. In another generated view, we perform the similar cluster-aware infomax learning and obtain the similar objective $\mathcal{L}_2$. The overall objective for the model is then combines two view's objectives together:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{18}$$

The training algorithm for unsupervised graph learning is summarized in Algorithm 1.

Algorithm 1: The training algorithm of CAMI.

Input: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}\}$, cluster number K, training epoch number E, hyperparameter parameters γ,λ
Output: Node embeddings $\mathbf{H}$

```
Generating two views G_1,G_2 of graph G;
for epoch ← 1, 2, ...E do
    X̃ ← shuffle(X);
    H ← f_θ(A, X),H̃ ← f_θ(A, X̃);
    s = Readout(H);
    z_k ← Clustering(H);
    Obtain L_1 and L_2 respect to each view using Eq. 17;
    Update f_θ by applying stochastic gradient descent using Eq. 18.
end
```

3.5. Computational Complexity Analysis.

The computational complexity of the CAMI mainly lies in obtaining node representations for two generated views and calculating multi-level mutual information estimation and maximization. When obtaining node representations for each view, we use the shared graph neural network encoder, so the complexity of the graph neural network is $O(l(Nd)^2)$, where l is the number of layers of GNN, N is the number of nodes of a graph, and d is the dimension of the node feature. When computing the mutual information estimation and maximization, the complexity for k-means clustering is $O(Nd'K)$ and the complexity for discriminator is $O(N^2d'K)$, where K is the number of clusters, d' is the dimension of high-level node representations. The overall computational complexity of the CAMI is $O(l(Nd)^2 + Nd'K + N^2d'K)$.

4. Experiments and Analysis

In this section, we conduct comprehensive tests to validate the model's effectiveness and examine the experimental data.

4.1. Experimental Setup

4.1.1. Datasets

We test CAMI's performance using seven regularly used real-world datasets. The statistics information is listed in Table 2 and brief information are as follows:

- **Amazon-Computers and Amazon-Photo** [34] are two networks sampled from Amazon's products. A node represents a product, and an edge is the co-purchase relationship, indicating two products are frequently bought together. The nodes' labels are the sub-categories under the category of "computer" and "photo". A product's review information is modeled as sparse bag-of-words features for a node.
- **Coauthor-CS and Coauthor-Physics** [34] are built on the co-authorship based on the Microsoft Academic Graph from the KDD Cup 2016 challenge. Each node indicates an author, and two nodes are linked if they occurred in a same publication. The author's paper keywords are used to encode the characteristic of a node. The label of a node represents the author's most active study field.
- **CORA and CiteSeer** [35] each with a node representing a paper-related machine learning subject and an edge reflecting the citation link between two articles. The attributed information is the description of the paper's abstract information. There are seven sub-categories in Cora and six sub-categories in CiteSeer that indicate the venue information of the paper.
- **PubMed** [36] is extracted and built based on PubMed database. Nodes in these networks are articles related to diabetes, and edges are the citation relationship. The node features are TF/IDF-weighted word frequencies and label specific to the type of diabetes addressed in the publication.

4.1.2. Baselines

We consider several representative graph learning methods for baselines:

- **Spectral Clustering** [37] is a method for addressing the partitioning of nodes in a graph. It is a softer variation of the conventional *k-means* clustering technique.
- **Node2vec** [19] uses Skip-Gram with negative sampling to learns continuous feature representations for nodes in networks.
- **DeepWalk** [20] learns the node embedding from a set of random walks. If nodes tend to co-occur on brief random walks throughout the network, they will have similar embeddings.
- **GAE and VGAE** [25] employ the encoder-decoder architecture to obtain node embeddings. The graph convolutional network serves as the encoder and an inner product serve as the decoder. VGAE is the non-probabilistic variant of GAE.
- **ARGA and ARVGA** [38] are adversarial graph embedding frameworks for graph data that train to reconstruct the graph structure based on the learned compact representation of nodes. ARGA and ARVGA are the two variants of adversarial approaches. The encoder uses both the topological structure and node content for learning.
- **DGI** [16] proposes a generic framework for learning graph node representations. It is based on unsupervised maximizing the mutual information between local patches and high-level graph summary representations.

Table 2
Statistics of datasets.

Datasets	#nodes	#edges	#attributes	#label
Amazon-Computers	13,753	245,861	767	10
Amazon-Photo	7650	119,081	745	8
Coauthor-CS	18,333	81,894	6805	15
Coauthor-Physics	34,493	247,962	8415	5
Cora	2708	5429	1433	7
CiteSeer	3327	4732	3703	6
PubMed	19,717	44,338	500	3

- **GIC** [39] is an unsupervised method that learns node representations and capture additionally cluster-level information content. Clusters are optimized by maximizing mutual information between cluster nodes.
- **MVGRL** [29] uses different graph diffusion kernels to augment the input graph and applies InfoMax principle in each view.
- **GRACE** [40] is a self-supervised method that generates two views of the graph and utilizes the contrastive objective at the node level to obtain the node representations. They also maximize the agreement of node representations from two different views.
- **MERIT** [41] obtains two contrastive views of the graph based on the local and global perspectives and utilizes the cross-view and cross-network objectives to train the model.
- **GCN** [7] is a semi-supervised model that learns graph embeddings based on aggregating and passing nodes' neighborhood information through a localized first-order approximation of spectral graph convolutions.
- **GAT** [6] applies the attention mechanism into GCN layers and specific the weights for each neighborhood when during the information passing process.

The approaches listed above may be classified into three types: (1) Traditional approaches: Spectral Clustering, Node2vec, and DeepWalk (2) Unsupervised approaches: GAE, VGAE, ARGA, ARVGA, DGI, GIC, MVGRL, GRACE, and MERIT; (3) Supervised approaches as counterparts: GCN and GAT.

4.1.3. Evaluation Metrics

There are different kinds of evaluation metrics for each evaluation task. Accuracy (ACC) is the commonly used metric for node classification problems. Based on the classification results, we have T_P, T_N, F_P and F_N represent the number of true positives, true negatives, false positives, and false negatives, respectively. The correctly classified percentage (ACC) is defined as:

$$ACC = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{19}$$

We use the AUC and AP as the performance metric for the link prediction task. AUC indicates the possibility that a randomly selected positive edge would rank higher than a randomly selected negative edge. The *Average Precision* score, abbreviated as AP, reflects the area under the precision-recall curve.

Precision and *Recall* are defined as:

$$Precision = \frac{T_P}{T_P + F_P} \tag{20}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{21}$$

For the clustering task, we use ACC, NMI, and ARI for evaluation. ACC is defined with Eq. 19. NMI is the normalized mutual information in the information theory that calculates depending on the mutual information I and the entropy of the labeled $H(Y)$ and the clustered sets $H(C)$:

$$NMI(Y, C) = \frac{2 \times I(Y, C)}{H(Y) + H(C)} \tag{22}$$

ARI is the Average Rand Index that represents an accuracy metric when additionally penalizing incorrect decisions. It is defined as:

$$ARI(P^*, P) = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{23}$$

where n_{ij}, a_i and b_j are values from the contingency table, which summarizes the overlapping clustering results. n_{ij} indicates the number of data points of the class label $C_j^* \in P^*$ assigned to cluster C_i in partition P, a_i indicates the number of data points in cluster C_i of partition P, and b_j indicates the number of data points in class C_j^*.

4.1.4. Implementation Details

The network encoder is a layer of GCN with a dimension size of 128 and activated with *pRelu* activation function. As the discriminator layer's scoring function, we utilize a trainable bilinear layer with logistic sigmoid nonlinearity. The readout function is an average operation on node representations and activated with logistic sigmoid nonlinearity. The learning rate of CAMI is initialized with $lr = 0.001$ and trained with maximum $E = 1000$ epochs using the Adam SGD optimizer. We use an early stop strategy if the accuracy has not been improved in 20 training steps and the training results show most experiments finish within 1000 epochs. In the structure-wise augmentation, the overall probability for an edge being removed is set to be $\alpha = 0.2$ and $\alpha = 0.3$, respectively. In the feature-wise augmentation, the overall probability of a feature dimension being masked is set to be $\beta = 0.1$. The number of clusters for the clustering component is configured to match the number of classes in the dataset, and the inverse-temperature hyperparameter is set to be $\gamma = 100$. The balance parameter λ to balance the adaptive multi-view generating's and the cluster-aware infomax learning's importance is initialized to be $\lambda = 0.5$. All the training and evaluating experiments are implemented in PyTorch and run on a Ubuntu 18.04 machine outfitted with four Nvidia GeForce RTX 2080 Ti GPUs.

4.2. Experimental Results

4.2.1. Clustering

The clustering task is an unsupervised task that aims to cluster unlabeled nodes into different categories based on the derived node embeddings. We cluster nodes into K categories using K-means clustering. As introduced above, we use accuracy (ACC), normalized mutual information (NMI), and average rand index

(ARI) as the metrics. Table 3 displays results of clustering for three datasets, and we can see that: (1) For all three measures, the CAMI consistently beats the baseline models in all three datasets. Compared to the most correlated DGI and GIC, the most significant improvement is a gain of 19.4% in PubMed for the ACC metric. (2) Although GIC is the most competitive method with an unsupervised clustering process in the training process, it still ignores the effect of noise edges and feature dimensions in a graph view. With the adaptive augmentation of the graph view that eliminates less important edges and feature dimensions, the multi-level mutual information estimation and maximization in the CAMI can learn from a more informative graph view and obtain inherent information of the dataset, thus benefiting the downstream clustering task.

4.2.2. Link Prediction

Some graph edges are missing in a link prediction task, and the aim is to forecast the chance of an edge forming between any two nodes. Following [25,38], the validation set has 5% edges and negative edges, the testing set has 10% edges and negative edges, and the final results are calculated by averaging the outcomes of ten runs. We use the ROC curve (AUC) and the average precision (AP) as the metric in the link prediction task. We report nine different models' performances on three datasets in Table 4. We make the following observation from Table 4: (1) In three datasets, the proposed CAMI obtains the highest AUC performance. Specifically, with AUC, CAMI outperforms GIC, the second best method, by around 3.1% in Cora, 1.3% in CiteSeer, and 3.4% in PubMed. As the AP metric, the CAMI obtains the best performance in Cora and CiteSeer and a slight disadvantage in PubMed. (2) GAE, VGAE, ARGA, and ARVGA are traditional GNN-based models and did not apply mutual information estimation in the training process. The CAMI achieves a mean AUC gain of more than 3.4%. We believe the mutual information estimate and maximization between a node's representation and the multi-levels of graph summaries have significantly promoted the node embedding and feature learning efficiency in an unsupervised manner. (3) Compared to DGI and GIC, that only use the InfoMax principle in a fixed graph view to estimate and maximize the mutual information, the proposed CAMI consistently improves the node representation with the adaptive view generating process and multi-level InfoMax learning and benefits the link prediction task.

4.2.3. Node Classification

The node classification task aims to predict a node's labels based on the given graph features, and the learned nodes' representations are passed through a classifier. We report different method's node classification results with seven datasets in Table 5. For each baseline method and dataset, following [34], the training set has 20 nodes sampled from each class, the validation set has 30 nodes sampled from each class, and the remaining nodes from each class to create the testing set. The final node classification accuracy (ACC) results are averaged over 20 times of test steps. The second column in Table 5 shows the available data for each method during training, where **X** is the node feature matrix, **A** is the adjacency matrix, and **Y** is the corresponding label. Besides the baselines introduced above, the "Row Feature" row represents the node classification results from logistic regression training on original input characteristics. The "DeepWalk + features" reports the results for DeepWalk with concatenated input features.

As we can observe from Table 5, the proposed CAMI obtains very competitive results over all seven datasets and consistently outperforms other unsupervised baselines. More specifically, we make the following observations: (1) Compared to the traditional deep walk-based methods (from the first row to the fourth row), our CAMI performs much better in all seven datasets and the improvements are significant, especially on Cora, CiteSeer, and PubMed, improving the node classification accuracy by 12.2%, 21.3%, and 6%, respectively. (2) The GNN-based unsupervised methods are the main comparison baselines from the fifth to the thirteenth row. It is noteworthy that the DGI, GIC, GMI, and MVGRL are methods that also use the Infomax principle to estimate and maximize the mutual information to learn the node embeddings. GRACE and MERIT are two self-supervised methods that generate two related views of the original graph and utilize the contrastive objective. The proposed CAMI achieves the best accuracy in five out of seven datasets, and in the remaining two datasets also performs very competitively, although we perform slightly inferior to the MERIT on the Core and Citeseer datasets. The MVGRL also generates different views of input graphs and employs diffusion to incorporate global information, which is the most similar to our method, it still performs slightly worse than our method in all seven datasets. This also illustrates that the adaptive multi-view augmentation carefully considers structure-wise centrality and feature-wise information that helps preserve the most informative graph feature in the generated view. (3) Compared to the supervised GCN and the GAT approaches that use label information during training, the proposed CAMI is an unsupervised learning model and only utilize the graph information itself. The CAMI achieves even higher accuracy in four datasets and performs slightly worse in the other three datasets, which proves the unsupervised graph generating and clustering in the CAMI framework have the ability to be adaptive to different datasets and improve the node representation learning quality.

In conclusion, compared to all baseline models, our approach achieves the best results in both clustering and link prediction tasks on the Cora, CiteSeer, and PubMed datasets. On the node classification task, the proposed CAMI has superior performance on all seven datasets, and the results of our model are very competitive even when compared with supervised methods. The results on the three tasks validates that the node representations learned through the CAMI framework are able to capture the informative multi-level graph structure information and feature attributes inherent in the dataset, and be beneficial for the downstream tasks.

Table 3
Clustering results on three datasets: ACC, NMI, and ARI.

	Cora			CiteSeer			PubMed		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Spectral Clustering	36.7	12.7	3.1	23.9	5.6	1.0	40.3	4.2	0.2
DeepWalk	48.4	32.7	24.3	33.7	8.8	9.2	68.4	27.9	29.9
GAE	59.6	42.9	34.7	40.8	17.6	12.4	67.2	27.7	27.9
VGAE	60.9	43.6	34.6	34.4	15.6	9.3	63.0	22.9	21.3
ARGA	64.0	44.9	35.2	57.3	35.0	34.1	66.8	30.5	29.5
ARVGA	63.8	45.0	37.4	54.4	26.1	24.5	69.0	29.0	30.6
DGI	59.0	38.6	33.6	57.9	30.9	27.9	49.9	15.1	14.5
GIC	72.5	53.7	50.8	69.6	45.3	46.5	67.3	31.9	29.1
CAMI(ours)	**73.1**	**55.6**	**52.0**	**70.5**	**45.7**	**46.9**	**69.3**	**32.6**	**32.0**

Table 4
Link prediction results on three datasets: AUC score and AP score (in %).

	Cora		CiteSeer		PubMed	
	AUC	AP	AUC	AP	AUC	AP
Spectral Clustering	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DeepWalk	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.4 ± 0.00	84.1 ± 0.00
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	96.4 ± 0.00	96.5 ± 0.00
VGAE	91.4 ± 0.01	92.6 ± 0.01	90.8 ± 0.02	92.0 ± 0.02	94.4 ± 0.02	94.7 ± 0.02
ARGA	92.4 ± 0.003	93.2 ± 0.003	91.9 ± 0.003	93.0 ± 0.003	96.8 ± 0.001	**97.1 ± 0.001**
ARVGA	92.4 ± 0.004	92.6 ± 0.004	92.4 ± 0.003	93.0 ± 0.003	96.5 ± 0.001	96.8 ± 0.001
DGI	89.9 ± 0.8	89.7 ± 1.0	95.5 ± 1.0	95.7 ± 1.0	91.2 ± 0.6	92.2 ± 0.5
GIC	93.5 ± 0.6	93.3 ± 0.7	97.0 ± 0.5	96.8 ± 0.5	93.7 ± 0.3	93.5 ± 0.3
CAMI(ours)	**96.6 ± 0.3**	**96.83 ± 0.4**	**98.3 ± 0.6**	**98.4 ± 0.4**	**97.1 ± 0.4**	97.0 ± 0.6

Table 5
Node classification accuracy (in %) of seven datasets.

Method	Training Data	Amazon-Computers	Amazon-Photo	Coauthor-CS	Coauthor-Physics	Cora	CiteSeer	PubMed
Raw Features	X	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00	47.9 ± 0.4	49.3 ± 0.2	69.1 ± 0.3
Node2vec	A	84.39 ± 0.08	89.67 ± 0.12	85.08 ± 0.03	91.19 ± 0.04	68.66 ± 1.83	47.98 ± 1.75	72.36 ± 0.95
DeepWalk	A	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15	67.2	43.2	65.3
DeepWalk + features	X,A	86.28 ± 0.07	90.05 ± 0.08	87.7 ± 0.04	94.90 ± 0.09	70.7 ± 0.6	51.4 ± 0.5	74.3 ± 0.9
GAE	X,A	85.27 ± 0.19	91.62 ± 0.13	90.01 ± 0.71	94.92 ± 0.07	70.6	53.1	72.5
VGAE	X,A	86.37 ± 0.21	92.2 ± 0.11	92.11 ± 0.09	94.52 ± 0.00	68.8	55.6	73.1
DGI	X,A	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.6
GIC	X,A	81.5 ± 1.0	90.4 ± 1.0	89.4 ± 0.4	93.1 ± 0.7	81.7 ± 1.5	71.9 ± 1.4	77.3 ± 1.9
GMI	X,A	82.21 ± 0.31	90.68 ± 0.17	OOM[1]	OOM[1]	82.7 ± 0.2	73.0 ± 0.3	80.1 ± 0.2
MVGRL	X,A	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03	82.9 ± 0.7	72.6 ± 0.7	79.4 ± 0.3
GRACE	X,A	-	81.8 ± 1.0	90.1 ± 0.8	-	80.0 ± 1.5	71.7 ± 0.6	79.5 ± 1.1
MERIT	X,A	-	87.4 ± 0.1	92.4 ± 0.4	-	**83.1 ± 0.4**	**74.0 ± 0.7**	80.1 ± 0.4
CAMI(ours)	X,A	**87.6 ± 0.4**	92.4 ± 0.2	**93.3 ± 0.2**	95.5 ± 0.24	82.9 ± 0.1	72.7 ± 0.2	**80.3 ± 0.3**
GCN	X,A,Y	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	**95.65 ± 0.16**	81.5	70.3	79
GAT	X,A,Y	86.93 ± 0.29	**92.56 ± 0.35**	92.31 ± 0.24	95.47 ± 0.15	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3

[1] OOM indicates Out-Of-Memory in the experiments.

4.3. Parameter Sensitivity

There are two hyperparameters in the proposed CAMI framework: the inverse-temperature hyperparameter γ in the clustering process and the balance parameter λ that controls the importance of two major components. In order to see these two hyperparameters' effects on the framework, we perform parameter sensitivity tests on three datasets. Table 6 displays the accuracy of node classification. We can see from Table 6, within the same value of λ, the varies of γ actually affect the node classification accuracy in different datasets, and different values of λ also perform inconsistently of the same γ value. For Cora, the best accuracy is obtained when $\lambda = 0.5$ and $\gamma = 100$, for CiteSeer, when $\lambda = 0.5$ and $\gamma = 150$ achieves the best result, and when $\lambda = 0.5$ and $\gamma = 150$ we obtain the highest accuracy. We can also see from Table 6 that both λ and γ should not be too small or too large. Otherwise, they would lead to a decline in the results, which means the adaptive multi-view generating component and the cluster-aware infomax learning component are the same important in the framework. Based on the sensitivity experiments and to obtain proper results in different datasets, we empirically set $\lambda = 0.5$ and $\gamma = 100$ in other experiments.

4.4. Ablation Study

We can see from Fig. 1 that the proposed CAMI contains two major components: the adaptive multi-view generating component that generates two correlated views and the cluster-aware infomax learning component that estimates and maximizes the mutual information between nodes and multi-level graph summaries. To verify each component's effectiveness, we conduct ablation study which contains three CAMI variants. Specifically, we have *CAMI-view* that excludes the multi-view generating process, *CAMI-graph* that excludes the graph-level mutual information estimation and maximization in each generated view, *CAMI-cluster* that excludes the cluster-level mutual information estimation and maximization in each generated view, and *CAMI* is the whole original model for comparison. Fig. 2 shows the node classification accuracy results of each variant model on three datasets. As we can see, each variation model's performance suffers to some extent, and the original CAMI consistently outperforms with different values of hyperparameter λ on different datasets. Fig. 2 proves that all framework components function as a complete model to accomplish graph representation learning and that none of them can be separated.

Table 6
Link Prediction ACC on Cora, CiteSeer, and PubMed.

	Cora				CiteSeer				PubMed			
$\gamma\lambda$	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
50	93.8	94.5	95.5	95.4	96.1	96.5	97.1	96.3	95.1	95.4	95.3	94.2
100	94.3	95.1	**96.6**	95.8	97.3	97.2	97.6	97.4	94.1	96.2	**97.1**	96.2
150	94.2	94.8	96.0	96.1	97.5	97.3	**98.3**	98.0	94.3	95.3	96.2	96.3
200	94.1	94.7	95.3	95.9	97.2	97.6	97.4	96.1	94.5	96.4	95.5	95.5

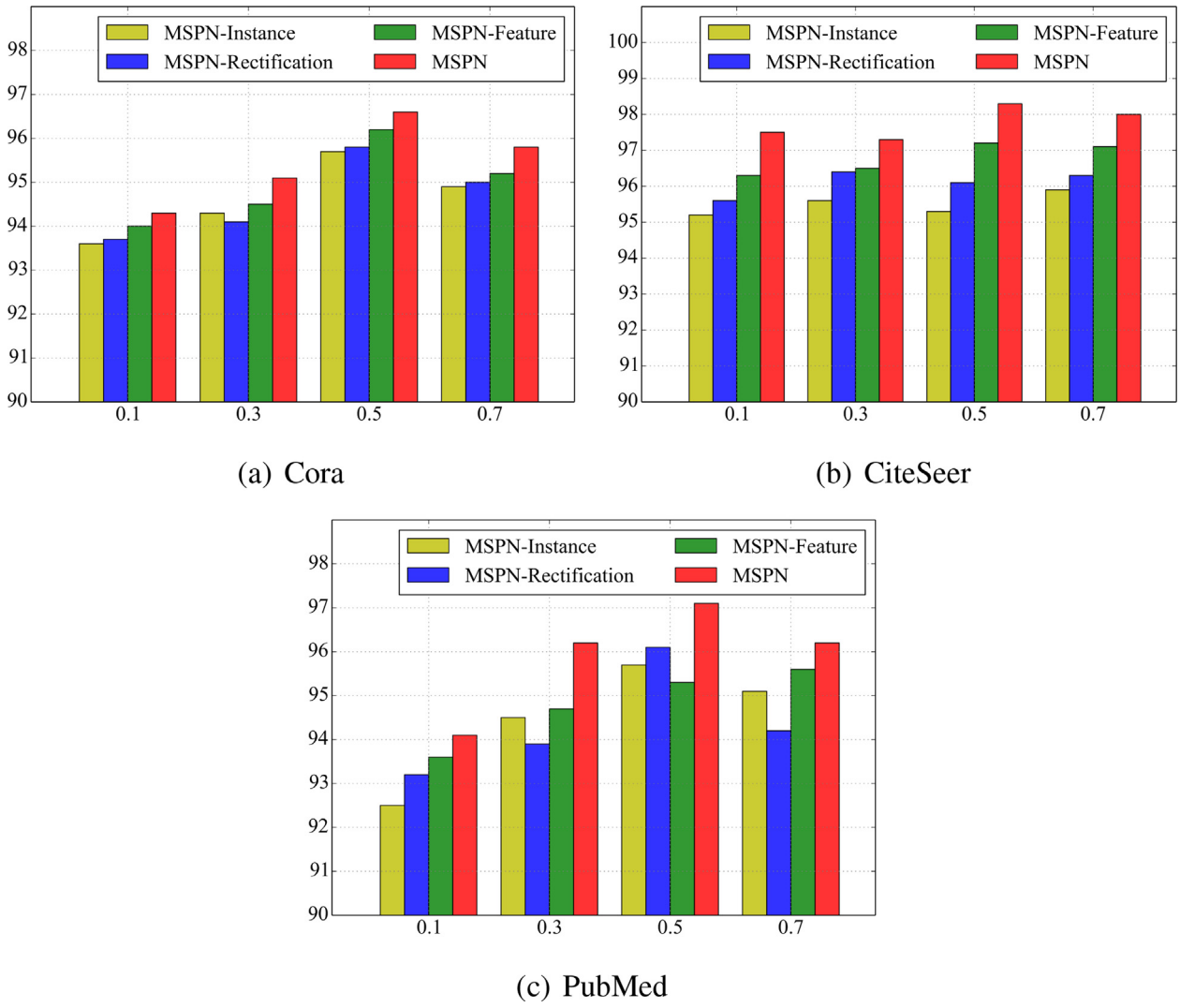

Fig. 2. Ablation study on three datasets.

5. Conclusion

In this paper, we propose Cluster-Aware Multiplex InfoMax (CAMI), an unsupervised graph representation learning architecture that augments two correlated views of the original graph in the first, and then applies multi-level mutual information maximization to better capture underlying topological information of the graph. The adaptive graph augmentation is performed on both the structure and node feature aspects. By identifying the importance of each node and each feature dimension, the augmentation scheme eliminates the unimportant edges and masks the noisy feature dimension to generate a view that contains the most informative topological structure and node features. In each view, we maximize the mutual information between a node's representation and a multi-level graph representation: the global graph-level summary representation and the cluster-level summary representation. The network encoder can capture richer structural content and improve the quality of node representations thanks to the multi-level mutual information maximization. The experimental results from seven real-world datasets indicate the efficiency of the learned node representations on different tasks, such as node classification, link prediction, and clustering tasks.

The proposed CAMI is a novel unsupervised graph learning framework which would be beneficial to the graph machine learning community. Our adaptive multi-view generating techniques help catching the inherent topological and node feature underlying the graph. And our cluster-aware infomax learning techniques encourage each generated node embedding containing multi-level graph information. When building graph-based applications, for example recommendation systems, these two modules will be useful for reducing the reliance on labeled data and improve the quality of the obtained node embeddings. In future work, we will carry out works on transferring the multi-view generating techniques and cluster-aware infomax learning techniques to unsupervised heterogeneous graph learning.

CRediT authorship contribution statement

Xin Xu: Methodology, Software, Writing - original draft. **Junping Du:** Writing - review & editing. **Jie Song:** Methodology, Software, Writing - original draft. **Zhe Xue:** Writing - review & editing. **Ang Li:** Validation, Visualization, Writing - review & editing. **Zeli Guan:** Validation, Visualization.

Data availability

Data will be made available on request.

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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62192784, 62172056, 62272058) and CCF-Tencent Open Fund (RAGR20220125).

References

[1] H. Li, P. Wu, Z. Wang, J. Mao, F.E. Alsaadi, N. Zeng, A generalized framework of feature learning enhanced convolutional neural network for pathology-image-oriented cancer diagnosis, Computers in Biology and Medicine 151 (2022).
[2] N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu, X. Liu, A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–14.
[3] H. Li, N. Zeng, P. Wu, K. Clawson, Cov-net: A computer-aided diagnosis method for recognizing covid-19 from chest x-ray images via machine vision, Expert Systems with Applications 207 (2022).
[4] D. Zeng, S. Zhao, J. Zhang, H. Liu, K. Li, Expression-tailored talking face generation with adaptive cross-modal weighting, Neurocomputing 511 (2022) 117–130, https://doi.org/10.1016/j.neucom.2022.09.025.
[5] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Advances in neural information processing systems 29.
[6] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, http://arXiv.org/abs/1710.10903.
[7] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, In Proceedings of the 6th International Conference on Learning Representations.
[8] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, J. Du, Leveraging graph to improve abstractive multi-document summarization, http://arXiv.org/abs/2005.10043.
[9] X. Li, W. Wei, X. Feng, X. Liu, Z. Zheng, Representation learning of graphs using graph convolutional multilayer networks based on motifs, Neurocomputing 464 (2021) 218–226, https://doi.org/10.1016/j.neucom.2021.08.028.
[10] F. Che, G. Yang, D. Zhang, J. Tao, T. Liu, Self-supervised graph representation learning via bootstrapping, Neurocomputing 456 (2021) 88–96, https://doi.org/10.1016/j.neucom.2021.03.123.
[11] D. Sun, D. Li, Z. Ding, X. Zhang, J. Tang, Dual-decoder graph autoencoder for unsupervised graph representation learning, Knowledge-Based Systems 234 (2021), https://doi.org/10.1016/j.knosys.2021.107564.
[12] C. Wang, X. Chen, B. Chen, F. Nie, B. Wang, Z. Ming, Learning unsupervised node representation from multi-view network, Information Sciences 579 (2021) 700–716, https://doi.org/10.1016/j.ins.2021.07.087.
[13] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural computation 7 (6) (1995) 1129–1159.
[14] M.I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, R.D. Hjelm, Mine: mutual information neural estimation, https://arxiv.org/abs/1801.04062.
[15] R.D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, Y. Bengio, Learning deep representations by mutual information estimation and maximization, https://arxiv.org/abs/1808.06670.
[16] P. Velickovic, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax, ICLR (Poster) 2 (3) (2019) 4.
[17] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, J. Huang, Graph representation learning via graphical mutual information maximization, in: Proceedings of The Web Conference 2020, 2020, pp. 259–270.
[18] S. Cao, W. Lu, Q. Xu, Grarep: Learning graph representations with global structural information, in: Proceedings of the 24th ACM international on conference on information and knowledge management, 2015, pp. 891–900.
[19] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.
[20] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701–710.
[21] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, J. Tang, Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec, in: Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 459–467.
[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proceedings of the 24th international conference on world wide web, 2015, pp. 1067–1077.
[23] K. Sun, Z. Zhu, Z. Lin, Adagcn: Adaboosting graph convolutional networks into deep models, https://arxiv.org/abs/1908.05081.
[24] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, https://arxiv.org/abs/1706.02216.
[25] T.N. Kipf, M. Welling, Variational graph auto-encoders, https://arxiv.org/abs/1611.07308.
[26] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, L. Van Gool, Scan: Learning to classify images without labels, in: European Conference on Computer Vision, Springer, 2020, pp. 268–285.
[27] C. Niu, H. Shan, G. Wang, Spice: Semantic pseudo-labeling for image clustering, https://arxiv.org/abs/2103.09382.
[28] Z. Xue, J. Du, H. Zhu, Z. Guan, Y. Long, Y. Zang, M. Liang, Robust diversified graph contrastive network for incomplete multi-view clustering, in: MM '22: The 30th ACM International Conference on Multimedia, 2022, ACM, 2022, pp. 3936–3944.
[29] K. Hassani, A.H. Khasahmadi, Contrastive multi-view representation learning on graphs, in: International Conference on Machine Learning, PMLR, 2020, pp. 4116–4126.
[30] T. Zhang, Y. Xiong, J. Zhang, Y. Zhang, Y. Jiao, Y. Zhu, Commdgi: community detection oriented deep graph infomax, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1843–1852.
[31] X. Liang, D. Li, A. Madden, Attributed network embedding based on mutual information estimation, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 835–844.
[32] M. Newman, Networks, Oxford University Press, 2018.
[33] B. Wilder, E. Ewing, B. Dilkina, M. Tambe, End to end learning and optimization on graphs, Advances in Neural Information Processing Systems 32.
[34] O. Shchur, M. Mumme, A. Bojchevski, S. Günnemann, Pitfalls of graph neural network evaluation, https://arxiv.org/abs/1811.05868.
[35] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI magazine 29 (3) (2008), 93–93.
[36] G. Namata, B. London, L. Getoor, B. Huang, U. EDU, Query-driven active surveying for collective classification, in: 10th International Workshop on Mining and Learning with Graphs, Vol. 8, 2012, p. 1.
[37] L. Tang, H. Liu, Leveraging social media networks for classification, Data Mining and Knowledge Discovery 23 (3) (2011) 447–478.
[38] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, Adversarially regularized graph autoencoder for graph embedding, In Proceeding of the Twenty-seven International Joint Conference on Artifical Intelligence, IJCAI-18 (2018) 2609–2615.
[39] C. Mavromatis, G. Karypis, Graph infoclust: Leveraging cluster-level node information for unsupervised graph representation learning, https://arxiv.org/abs/2009.06946.
[40] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Deep graph contrastive representation learning, https://arxiv.org/abs/2006.04131.
[41] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, S. Pan, Multi-scale contrastive siamese networks for self-supervised graph representation learning, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), 2021, pp. 1477–1483.

Xin Xu is currently working toward the Ph.D degree with Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include machine learning, intelligent information processing and knowledge graph.

Junping Du is now a Professor and PhD tutor at the School of Computer Science and Technology, Beijing University of Posts and Telecommunications, China. Her research interests include artificial intelligence, machine learning and pattern recognition.

Jie Song was born in 1997. He received the master's degree from Beijing University of Posts and Telecommunications in 2022. He now works at FreeWheel. His main research interests include data mining, information retrieval and machine learning.

Zhe Xue received the PhD degree in computer science from University of Chinese Academy of Sciences, China in 2017. He is currently an Associate Professor with the school of computer science, Beijing University of Posts and Telecommunications, China. His research interests include machine learning, data mining and multimedia data analysis.

Ang Li is a Ph.D. candidate in Computer Science and Technology at the Beijing University of Posts and Telecommunications. He received a B.S. degree from the Nanchang Hangkong University in 2015 and an M.S. degree from the Beijing University of Posts and Telecommunications in 2019, all related to computer science. His research interests include information retrieval, scholar profiling, and data mining.

Zeli Guan is an ph.D. candidate of School of Computer Science, Beijing University of Posts and Telecommunications. His mainly research directions for federated Learning, graph neural network and machine learning. His research papers have been published in or accepted by International Journal of Intelligent Systems, Computational Intelligence and Neuroscience, and Visual Informatics.