Cross-View Graph Consistency Learning for Invariant Graph Representations

Jie Chen[1, 4], **Hua Mao**[2], **Wai Lok Woo**[2], **Chuanbin Liu**,[3*], **Xi Peng**[1]

[1]College of Computer Science, Sichuan University
[2]Department of Computer and Information Sciences, Northumbria University
[3]Center for Scientific Research and Development in Higher Education Institutes, Ministry of Education
[4]National Key Laboratory of Fundamental Algorithms and Models for Engineering Numerical Simulation, Sichuan University
chenjie2010@scu.edu.cn, {hua.mao, wailok.woo}@northumbria.ac.uk, liuchuanbin@cutech.edu.cn, pengx.gm@gmail.com

Abstract

Graph representation learning is fundamental for analyzing graph-structured data. Exploring invariant graph representations remains a challenge for most existing graph representation learning methods. In this paper, we propose a cross-view graph consistency learning (CGCL) method that learns invariant graph representations for link prediction. First, two complementary augmented views are derived from an incomplete graph structure through a coupled graph structure augmentation scheme. This augmentation scheme mitigates the potential information loss that is commonly associated with various data augmentation techniques involving raw graph data, such as edge perturbation, node removal, and attribute masking. Second, we propose a CGCL model that can learn invariant graph representations. A cross-view training scheme is proposed to train the proposed CGCL model. This scheme attempts to maximize the consistency information between one augmented view and the graph structure reconstructed from the other augmented view. Furthermore, we offer a comprehensive theoretical CGCL analysis. This paper empirically and experimentally demonstrates the effectiveness of the proposed CGCL method, achieving competitive results on graph datasets in comparisons with several state-of-the-art algorithms.

Introduction

Graph neural networks (GNNs), inheriting the representation learning advantages of traditional deep neural networks (Wang et al. 2023; Liu et al. 2023a; Chen et al. 2023a,c; Peng et al. 2022), have become increasingly popular for analyzing graph-structured data (Hassani and Khasahmadi 2020). Graph representation learning (GRL) aims to learn meaningful node embeddings, referred to as graph representations, from both graph structures and node features via GNNs (Kipf and Welling 2017). Graph representations have been extensively applied in downstream tasks, e.g., link prediction. Link prediction seeks to predict the missing connections between node pairs from an incomplete graph structure. It shows the significant impact of developing applications with graph-structured data (Liu et al. 2023b), including citation network analysis (Li et al. 2022a), social network analysis (Li et al. 2022a) and recommendation systems (Wu et al. 2021).

In recent years, many research efforts have been directed toward investigating GNNs for GRL. For example, several classic GNN models have been proposed to learn meaningful graph representations from graph-structured data, such as graph convolutional network (GCN) (Kipf and Welling 2017), variational graph autoencoder (VGAE) (Kipf and Welling 2016), graph attention network (Veličkovi'c et al. 2018), graph sampling and aggregation (GraphSAGE) (Hamilton, Ying, and Leskovec 2017) and their variants (Tan et al. 2023; Li et al. 2022a). These methods have achieved impressive link prediction results. However, the potential of utilizing graph structures for the available graph-structured data has not been fully exploited.

Self-supervised learning has recently emerged as a promising representation learning paradigm for GNNs (Hou et al. 2022; Tsai et al. 2020). It can learn latent graph representations from unlabeled graph-structured data by supervision, which is provided by the data itself with different auxiliary learning tasks. Most self-supervised learning-based algorithms fall into two categories: contrastive learning (Chen et al. 2023b; Li et al. 2022b; Hassani and Khasahmadi 2020; You et al. 2020) and generative learning (Tan et al. 2023; Li et al. 2022a; Hou et al. 2022; Cui et al. 2020). For example, Hassani *et al.* (Hassani and Khasahmadi 2020) presented a contrastive multiview representation learning (CMRL) method that learns graph representations by contrasting encodings derived from first-order neighbors and a graph diffusion module. The feature pairs may have different data distributions under the two different types of augmented views. This may have a significant negative impact on measuring the similarity between positive pairs and the dissimilarity between negative pairs when conducting contrastive learning. Li *et al.* (Li et al. 2022a) presented a masked graph autoencoder (MGAE) method that reconstructs a complete graph structure by masking a portion of the observed edges. By randomly masking a portion of the edges, the MGAE method somewhat reduces the redundancy of the graph autoencoder (GAE) in self-supervised graph learning tasks. The randomness of edge masking causes a sampling information loss problem. Consequently, this is an increasing concern regarding the capture of the complementary information between the augmented views of a graph.

[*]Corresponding author.

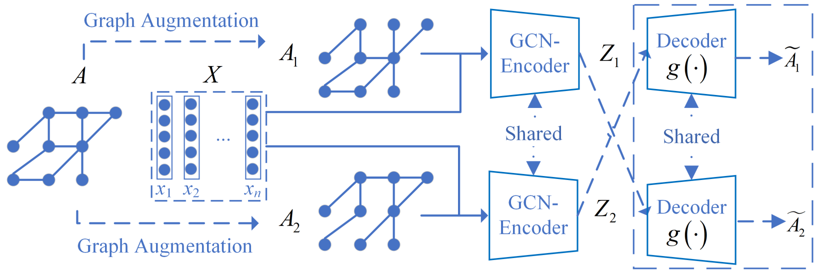


Figure 1: Framework of the CGCL model. Each augmented view of a graph structure, $\mathbf{A}_i$ $(i = 1, 2)$, corresponds to three modules, including an augmented graph structure module, a shared GCN encoder module and a shared cross-view consistency decoder module. $\mathbf{A}$ and $\widetilde{\mathbf{A}}$ represent the incomplete graph structure and predictive graph structure, respectively.

A vast majority of the available contrastive learning-based GRL methods consider their GNN-based models to be independent of different downstream tasks (Tan et al. 2023; Zhu et al. 2020). Recent advances have shown that the optimal data augmentation views critically depend on the downstream tasks involving the visual data (Wang et al. 2022; You et al. 2020). The augmented views of the visual data share as little information as necessary to maximize the task-relevant mutual information. Motivated by this, we further investigate how to take full advantage of graph structure information and simultaneously retain the task-relevant information needed by a specific downstream task, i.e., link prediction. This is beneficial for learning invariant graph representations from the augmented views of graph-structured data, which is central to enhancing the ability of a model to produce general graph representations.

In this paper, we propose a cross-view graph consistency learning (CGCL) method that learns invariant graph representations for link prediction. Numerous missing connections are involved in the graph structure. We first construct two complementary augmented views of the graph structure of interest from the remainder of the link connections. Then, we propose a cross-view training scheme to train the proposed CGCL model, which can produce invariant graph representations for graph structure reconstruction purposes. The proposed cross-view training scheme attempts to maximize the consistency information between one augmented view and the graph structure reconstructed from the other augmented view. In contrast with random edge perturbation, node dropping or attribute masking, this approach preserves the valuable information contained in raw graph-structured data during GRL. Moreover, the complementary information in the graph structure can be further exploited by virtue of the cross-view training scheme. In addition, a comprehensive theoretical analysis is provided to reveal the supervisory information connections between the two complementary augmented views.

The key contributions are summarized as follows.

- We propose a CGCL model that learns two pairs of cross-view adjacency matrices, which can be considered view-invariant, for link prediction.
- A coupled graph structure augmentation scheme is introduced to construct two complementary augmented views of a graph structure, which help produce more distinct low-dimensional node embeddings.
- A cross-view training scheme is designed to train the CGCL model by maximizing the consistency information between one augmented view and the graph structure reconstructed from the other augmented view.
- Extensive experiments are conducted on graph datasets, achieving competitive results.

Preliminaries

An undirected graph $\mathcal{G}$ is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_n\}$ and $\mathcal{E} = \{e_1, ..., e_m\}$ stand for a set of n nodes and a set of m edges, respectively (Kipf and Welling 2017). Each node v_i in $\mathcal{V}$ is associated with a corresponding feature x_i $(1 \leq i \leq n) \in \mathbb{R}^d$. An adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ represents the relationships among the nodes in a graph, where $A_{ij} = 1$ indicates that an edge exists between nodes v_i and v_j and vice versa. The degree matrix $\mathbf{D}$ is defined as $\mathbf{D} = diag\left[d_1, d_2, ..., d_n\right] \in \mathbb{R}^{n \times n}$, and its diagonal elements are $d_i = \sum_{v_j \in V} A_{ij}$.

A GCN learns node embeddings for graph-structured data (Kipf and Welling 2017). Given an undirected graph $\mathcal{G}$, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix of $\mathcal{G}$ with an added self-loop, and $\widetilde{\mathbf{D}}$ is a diagonal degree matrix $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. The formula for the lth GCN layer is defined as

$$\mathbf{H}^{(l)} = \sigma\left(\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}\right) \tag{1}$$

where l denotes the lth layer, $\mathbf{W}^{(l)}$ is a layer-specific learnable weight matrix, $\mathbf{H}^{(l)}$ is the node embedding matrix with $\mathbf{H}^{(0)} = \mathbf{X}$, and σ is a nonlinear activation function, e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. For a semi-supervised classification task, the weight parameters in the GCN model can be learned by minimizing the cross-entropy error between the ground truth and predictive the labels (Kipf and Welling 2017).

Cross-View Graph Consistency Learning

In this section, we present the proposed CGCL method in detail, which produces invariant graph representations for self-supervised graph learning. These graph representations can be employed on a specific downstream task, i.e., the reconstruction of an incomplete graph structure.

Problem Formulation

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a matrix consisting of node features, each row of which corresponds to a node feature. Given an undirected graph $\mathcal{G}$ with an incomplete graph structure and the above node features $\mathbf{X}$, the goal of our work is to learn graph representations, which can be employed to reconstruct the incomplete graph structure for link prediction.

Network Architecture

The proposed CGCL method aims to produce the complete graph structure by predicting the missing connections. Fig. 1 provides an overview of the proposed CGCL network architecture, which is composed of three main modules, i.e., an augmented graph structure module, a shared GCN encoder module and a shared cross-view consistency decoder module. The augmented graph structure module is utilized to generate two complementary augmented views of the original graph structure. Inspired by the VGAE (Kipf and Welling 2016), the shared GCN encoder module learns individual graph structures for each augmented view under unsupervised representation learning. The cross-view consistency decoder module produces a predictive graph structure by maximizing the consistency information between one augmented view and the graph structure reconstructed from the other augmented view. With these three modules, CGCL simultaneously learns graph representations from graph-structured data and reconstructs an incomplete graph structure for inferring the missing connections.

CGCL Model

Motivation Considering the two complementary graph structures $\mathbf{A}_1$ and $\mathbf{A}_2$ shown in Fig. 1, CGCL aims to ensure the consistency of the pairwise matchings between the pairs of nodes in $\mathbf{A}_1$ and $\widetilde{\mathbf{A}_1}$, where $\widetilde{\mathbf{A}_1}$ is produced from the other original structure $\mathbf{A}_2$ and the node features.

Definition 1 (Cross-View Graph Consistency) *Given two augmented graph structures $\mathbf{A}_1$ and $\mathbf{A}_2$, a generated graph structure $\widetilde{\mathbf{A}_1}$ is assumed to be reconstructed by using the node features $\mathbf{X}$ and graph structure $\mathbf{A}_2$. Let A^1_{ij} and $\widetilde{A^1_{ij}}$ $(i, j \in \{1, ..., n\})$ form a pairwise matching chosen from $\mathbf{A}_1$ and $\widetilde{\mathbf{A}_1}$, respectively. The relationship between the two graph structures $\mathbf{A}_1$ and $\widetilde{\mathbf{A}_1}$ is said to exhibit cross-view consistency if the following equality holds: $A^1_{ij} = \widetilde{A^1_{ij}}$ for all $i, j \in \{1, ..., n\}$.*

To reconstruct the incomplete graph structure, CGCL maximizes the consistency information between two random variables v_1 and v_2 with a joint distribution $p(v_1, v_2)$ corresponding to the pairwise matching variables of a pair of nodes a_1 and a_2 in a graph structure $\mathbf{A}_1$ and its reconstruction $\widetilde{\mathbf{A}_1}$, i.e.,

$$\max_f C(a_1, a_2) \tag{2}$$

where $C(a_1, a_2) = \mathbb{E}_{p_{(a_1,a_2)}} a_1 \log a_2$, $a_i = f(v_i)$ denote random variables, $v_i \geq 0, i \in \{1, 2\}$, and f represents a mapping function. According to the data processing inequality for the Markov chains $v_1 \rightarrow v_2 \rightarrow a_2$ and $a_2 \rightarrow v_1 \rightarrow a_1$ (Wang et al. 2022; Cover and Thomas 2006), we have

$$C(v_1, v_2) \geq C(v_1, a_2) \geq C(a_1, a_2). \tag{3}$$

Thus, $C(v_1, v_2)$ is the upper bound of $C(a_1, a_2)$. The variable v_1 can provide supervisory information for v_2 in an unsupervised manner, and vice versa. All supervisory information for one augmented view comes from the consistency information provided by the other augmented view in the context of CGCL. Assuming that the mapping function f has a sufficient graph representation learning ability, we have $C(v_1, v_2) = C(a_1, a_2)$. This indicates that $C(v_1, v_2)$ and $C(a_1, a_2)$ are approximately minimal when the consistency information shared between the two augmented views is sufficient for each other during GRL.

Coupled Graph Structure Augmentation The hidden representation of each node during GRL is determined by both the node itself and its neighbors. The neighbor selection process is influenced by both the graph structure and the node features. To extract sufficient supervisory information from the two augmented views, the straightforward approach is to select distinct neighbor candidate sets for each node. In contrast to the recently proposed graph contrastive learning methods that employ techniques such as node dropping and attribute masking for graph data augmentation, we refrain from performing any augmentation operations on the node features of the graph.

Guided by the task of reconstructing the incomplete graph structure, a portion of the connections is missing in an undirected graph $\mathcal{G}$. Edge dropping is an intuitive graph-structured data augmentation. The choice of different edge dropping ratios often leads to varying results in link prediction tasks. Some combinations of dropping ratios can even perform worse than no augmentation at all. Consequently, determining an optimal dropping ratio for edges becomes a dilemma. We introduce a coupled graph structure augmentation scheme for the original incomplete graph structure. Specifically, we randomly divide the set of edges into two complementary subsets following a particular distribution, e.g., the Bernoulli distribution. The edges of the two subsets are complementary when the directions of the edges are not considered. Furthermore, two undirected augmented views $\mathbf{A}_1$ and $\mathbf{A}_2$ can be constructed after applying the bidirectional order for each pair of nodes in each subset, which are $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively. In particular, these two views may share some edge pairs. This strategy offers two benefits for graph-structured augmentation. First, it directly simplifies the process of selecting the optimal edge-dropping ratio. Second, graph representation learning models with multiple GNN layers tend to make node representations more similar due to their message-passing mechanisms. By providing complementary candidate neighbor sets for each node, these two augmented views help produce more distinct low-dimensional node embeddings.

Given the adjacent matrices of the two complementary augmented views for a graph structure, $\mathbf{A}_1$ and $\mathbf{A}_2$, two corresponding adjacency matrices can be reconstructed using the GNN-based models (Veličkovi'c et al. 2018; Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2017). Different from previous work (Tan et al. 2023; Fang et al. 2022; Li et al. 2022a), we focus on reconstructing the cross-view adjacency matrices from $\mathbf{A}_1$ and $\mathbf{A}_2$ based on our specific motivation. Specifically, the cross-view adjacency matrices $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$ can be reconstructed as follows:

$$\begin{aligned} \widetilde{\mathbf{A}}_1 &= f\left(\mathbf{X}, \mathbf{A}_2\right), \\ \widetilde{\mathbf{A}}_2 &= f\left(\mathbf{X}, \mathbf{A}_1\right) \end{aligned} \tag{4}$$

where f is a mapping function consisting of the shared GCN encoder module and shared cross-view consistency decoder module shown in Fig. 1. $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$ represent the prediction results obtained for the incomplete graph structures derived from $\mathbf{A}_2$ and $\mathbf{A}_1$, respectively. Two pairs of adjacency matrices produced from graph-structured data, $\left(\mathbf{A}_1, \widetilde{\mathbf{A}}_1\right)$ and $\left(\mathbf{A}_2, \widetilde{\mathbf{A}}_2\right)$, are cross-view adjacency matrices that can be considered view-invariant matrices. These pairs of adjacency matrices can be utilized for link prediction.

Reconstruction of an Incomplete Graph Structure The shared GCN encoder module utilizes a GCN as the backbone. For each GCN-based encoder component, the encoder part produces hidden representations of the nodes derived from a graph structure and node features. Without a loss of generality, the GCN-based encoder component is defined as

$$\mathbf{Z}_v = g\left(\mathbf{X}, \mathbf{A}_v\right) = \mathrm{ELU}\left(\mathbf{A}_v \mathbf{X} \mathbf{W}^{(1)}\right), \tag{5}$$

where $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times d_v}$ is the weight parameters, d_v represents the number of neural units in the hidden layer, $\mathrm{ELU}\left(\cdot\right)$ denotes a nonlinear activation function, and $\mathbf{Z}_v$ represents low-dimensional node embeddings. Thus, the GCN-based encoder component extracts graph-level representations $\mathbf{Z}_1$ and $\mathbf{Z}_2$ for the adjacent matrices of the two augmented views $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively.

Given the low-dimensional node embeddings $\mathbf{Z}_v$, we employ the shared cross-view consistency decoder module, which consists of an inner product decoder component and a multilayer perceptron (MLP) component, to obtain the reconstructed adjacency matrices. The inner product decoder component is constructed by

$$\mathbf{H}_v = \sum_{i=1}^{N} \sum_{j=1}^{N} p\left(\mathbf{A}_{ij}^{(v)} | \mathbf{Z}_i^{(v)}, \mathbf{Z}_j^{(v)}\right) \tag{6}$$

where $p\left(\mathbf{A}_{ij}^{(v)} | \mathbf{Z}_i{}^{(v)}, \mathbf{Z}_j{}^{(v)}\right) = \left(\mathbf{Z}_i^{(v)}\right)^T \mathbf{Z}_j^{(v)}$ and $v \in \{1, 2\}$. The MLP component is composed of two fully connected layers with computations defined as follows:

$$\begin{aligned} \widetilde{\mathbf{A}}_1 &= \mathbf{MLP}\left(\mathbf{H}_2, \mathbf{W}^{(1,2)}\right), \\ \widetilde{\mathbf{A}}_2 &= \mathbf{MLP}\left(\mathbf{H}_1, \mathbf{W}^{(1,2)}\right) \end{aligned} \tag{7}$$

where a nonlinear activation function, $\mathrm{ReLU}\left(\cdot\right)$, is applied for each layer. $\mathbf{H}_1$ and $\mathbf{H}_2$ represent invariant graph representations that are utilized to predict $\widetilde{\mathbf{A}}_2$ and $\widetilde{\mathbf{A}}_1$, respectively. The reconstructed graph structures $\widetilde{\mathbf{A}_1}$ and $\widetilde{\mathbf{A}_2}$ can be obtained using Eq. (7).

Training Scheme

We propose a cross-view scheme for training the CGCL model. For the connections between the same pair of nodes, we emphasis their consistency across the augmented view and its corresponding counterpart. Specifically, the difference between the adjacent matrices of the augmented view $\mathbf{A}_v$ and the reconstructed graph structure $\widetilde{\mathbf{A}_v}$ should be reduced during the training stage ($v \in \{1, 2\}$). We establish two sets of edges to measure this difference. First, we create a set of edges by selecting all edges from $\mathbf{A}_v$ ($v \in \{1, 2\}$), considering each edge as positive. Second, we generate a set of negative edges by randomly sampling unconnected edges from the alternate augmented view. The number of negative edges corresponds to the number of positive edges within the alternative augmented view. By adopting the binary cross-entropy loss, the loss function for the graph consistency L_v between $\mathbf{A}_v$ and $\widetilde{\mathbf{A}_v}$ is formulated as:

$$\begin{aligned} L_v &= loss\left(\mathbf{A}_v, \widetilde{\mathbf{A}_v}\right) \\ &= -\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\delta\left(\mathbf{A}_{ij}^{v}\right) \log \delta\left(\widetilde{\mathbf{A}_{ij}^{v}}\right)\right. \\ &\quad \left. + \left(1 - \delta\left(\mathbf{A}_{ij}^{v}\right)\right) \log\left(1 - \delta\left(\widetilde{\mathbf{A}_{ij}^{v}}\right)\right)\right] \end{aligned} \tag{8}$$

where $\mathbf{A}_{ij}^{v}$ and $\widetilde{\mathbf{A}_{ij}^{v}}$ represent the elements of the ith rows and the jth columns of $\mathbf{A}_v$ and $\widetilde{\mathbf{A}_v}$, respectively. The entire learning procedure of the proposed CGCL method is summarized in Algorithm 1. Thus, we can obtain the graph representations and the predictive graph structure, simultaneously.

Theoretical Analysis

In this section, we provide a theoretical analysis of our model from the perspective of sufficient supervision information in GRL. The reconstructions of the incomplete graph structures $\widetilde{\mathbf{A}_1}$ and $\widetilde{\mathbf{A}_2}$ are obtained from the two augmented views. Considering Eq. (8), the general form of the optimization objective is

$$loss\left(\mathbf{A}_v, \widetilde{\mathbf{A}_v}\right) \tag{9}$$

where $v \in \{1, 2\}$. According to Eq. (3), a loss of task-relevant information occurs if the following condition holds, i.e.,

$$C\left(\mathbf{A}_1, \mathbf{A}_2\right) \geq C\left(\widetilde{\mathbf{A}_1}, \widetilde{\mathbf{A}_2}\right). \tag{10}$$

Similarly, the sufficient supervisory information shared between one augmented view and the graph structure reconstructed from the other augmented view is task-relevant if the following condition holds, i.e.,

$$C\left(\mathbf{A}_1, \widetilde{\mathbf{A}_1}\right) = C\left(\mathbf{A}_2, \widetilde{\mathbf{A}_2}\right) = C\left(\widetilde{\mathbf{A}_1}, \widetilde{\mathbf{A}_2}\right). \tag{11}$$

By considering the objective from the perspective of sufficient supervisory information, we emphasize the importance of utilizing appropriate data augmentation schemes for incomplete graph structures. Therefore, the augmented view

Algorithm 1: Optimization Procedure for CGCL

```
Input: Data matrix $\mathbf{X}$, the adjacency matrix of an incomplete graph structure $\mathbf{A}$, and parameters $\lambda$ and $d_v$.
Initialize: $epochs = 800$;
for $t = 1$ to $epochs$ do
    Constructing two complementary augmented views $\mathbf{A_1}$ and $\mathbf{A_2}$ from $\mathbf{A}$;
    for $v = 1$ to 2 do
        $i = 1$ if $v == 2$ else 2;
        Computing $\mathbf{Z}^{(v)}$ via Eq. (5) using $\mathbf{X}$ and $\mathbf{A}_v$;
        Constructing the set $\mathbf{E}_i^{'}$ by sampling the negative edges from the other augmented view $\mathbf{A}_i$;
        Computing $\mathbf{H}^{(v)}$ via Eq. (6) using three variables, including $\mathbf{Z}^{(v)}$, the set of edges constructed from $\mathbf{A}_i$, and the set of negative edges $\mathbf{E}_i^{'}$;
        Computing $\widetilde{\mathbf{A}}_i$ via Eq. (7);
        Updating $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(1,2)}$ by minimizing $L_v$ in Eq. (8) using $\mathbf{A}_i$ and $\widetilde{\mathbf{A}}_i$;
    end for
end for
Computing $\mathbf{Z}$ via Eq. (5) using $\mathbf{X}$ and $\mathbf{A}$;
Computing $\mathbf{H}$ via Eq. (6);
Computing $\widetilde{\mathbf{A}}$ via Eq. (7);
Output: The graph structure $\widetilde{\mathbf{A}}$.
```

generation process can typically be guided by the theoretical conditions in (10) and (11).

Theorem 1 *For any two variables $r_1 \in [0,1]$ and $r_2 \in [0,1]$, $C(a_1, a_2)$ is bounded by:*

$$-\frac{\log 2}{1+1/e} \leq C(a_1, a_2) \leq -\frac{\log(1+1/e)}{2}$$

where $a_1 = \delta(r_1)$, $a_2 = \delta(r_2)$, and $\delta(\cdot)$ denotes the sigmoid function.

Proof *Let $r_i \in [0,1]$ $(i \in \{1,2\})$ be any two variables. According to the sigmoid function $\delta(\cdot)$, we have*

$$\frac{1}{2} \leq a_i \leq \frac{1}{1+1/e}.$$

Considering $C(a_1, a_2) = \mathbb{E}_{p(a_1, a_2)} a_1 \log a_2$, we obtain

$$-\frac{\log 2}{1+1/e} \leq C(a_1, a_2) \leq -\frac{\log(1+1/e)}{2}.$$

□

According to Theorem 1, L_v has specific upper and lower bounds in Eq. (8). The lower bound can be theoretically guaranteed when minimizing L_v in Eq. (8). Moreover, we further obtain the following Lemma.

Lemma 1 *There exists a constant such that $\forall a_1, a_2$ and $\forall \theta_1, \theta_2 \in \theta$, the following inequality holds:*

$$|C_{\theta_1}(a_1, a_2) - C_{\theta_2}(a_1, a_2)| \leq \frac{1}{1+1/e}$$

where $\theta \in \mathbb{R}^d$ represents the parameters in neural networks.

The proof of Lemma 1 is omitted, as it follows in a straightforward manner from Theorem 1. Lemma 1 predicts that the loss function L_v will gradually decline during the training stage.

Related Work

Contrastive learning-based GRL methods follow the principle of mutual information maximization by contrasting positive and negative pairs (You et al. 2020; Hassani and Khasahmadi 2020; Zhu et al. 2020). Data augmentation is a key prerequisite for these GRL methods. For example, You *et al.* (You et al. 2020) presented a graph-contrastive learning framework that provides four types of graph augmentation strategies, including node dropping, edge perturbation, attribute masking and subgraph production, to improve the generalizability of the graph representations produced during GNN pretraining.

Generative learning-based GRL methods aim to reconstruct graph data for learning graph representations (Tan et al. 2023; Li et al. 2022a; Kipf and Welling 2016). For example, a VGAE employs two GCN models to build its encoder component, which can learn meaningful node embeddings for the reconstruction of a graph structure (Kipf and Welling 2016). Lousi *et al.* (Louis, Jacob, and Salehi-Abari 2023) presented a scalable simplified subgraph representation learning method (S3GRL) that simplifies the message passing and aggregation operations in the subgraph of each link. Additionally, several GAE-based GRL methods employ different masking strategies on a graph structure to implement graph augmentation, including a masked GAE (GraphMAE) (Hou et al. 2022), an MGAE (Li et al. 2022a) and a self-supervised GAE (S2GAE) (Tan et al. 2023). After randomly making a portion of the edges on a graph structure, these GRL methods attempt to reconstruct the missing connections with a partially visible, unmasked graph structure.

Experiments

In this section, we conduct extensive experiments to evaluate the link prediction performance of the proposed CGCL method. The source code for CGCL is implemented upon a PyTorch framework and a PyTorch Geometric (PyG) library. All experiments are performed on a Linux workstation with a GeForce RTX 4090 GPU (24-GB caches), an Intel (R) Xeon (R) Platinum 8336C CPU and 128.0 GB of RAM.

Experimental Settings

Datasets We select five widely used graph datasets for evaluation, including Cora (Sen et al. 2008), Citeseer (Sen et al. 2008), Pubmed (Namata et al. 2012), Photo (McAuley et al. 2015), and Computers (McAuley et al. 2015), which are publicly available on PyG. The Cora, Citeseer and Pubmed datasets are citation networks, where nodes and edges indicate papers and citations, respectively. The Photo and Computers datasets are segments of the Amazon copurchase graph, where each node represents a good, and each edge indicates that the two corresponding goods are frequently bought together.

Methods	Testing ratios	Cora		CiteSeer		PubMed		Photo		Computers	
		AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
GraphSAGE	10%	90.56±0.39	88.70±0.62	89.22±0.47	88.15±0.67	94.85±0.10	94.97±0.08	90.42±1.35	87.99±1.27	87.20±1.38	84.99±1.18
VGAE		91.88±0.58	92.51±0.35	91.29±0.56	92.35±0.41	94.86±0.69	94.75±0.62	96.76±0.16	96.72±0.16	90.86±0.69	91.75±0.62
SEAL		91.11±0.13	92.87±0.15	89.08±0.15	91.44±0.05	93.18±0.17	93.84±0.07	96.46±0.14	97.11±0.13	94.53±0.16	94.69±0.31
S3GRL		94.24±0.22	94.04±0.54	95.79±0.63	95.04±0.68	97.12±0.29	96.72±0.40	96.90±0.52	96.44±0.43	95.17±0.45	93.08±0.65
S2GAE		95.89±0.48	95.78±0.60	95.65±0.26	95.75±0.33	96.85±0.13	96.49±0.13	97.93±0.06	97.39±0.09	97.25±0.17	96.68±0.21
MGAE		96.74 ±0.09	96.36 ±0.18	**97.62±0.13**	**97.87±0.13**	97.62±0.02	97.19±0.04	98.64±0.01	98.42±0.02	98.27±0.03	98.01±0.03
CGCL		**97.00±0.15**	**97.34±0.11**	97.35±0.23	97.62±0.16	**98.48±0.03**	**98.37±0.03**	**98.88±0.01**	**98.72±0.02**	**98.41±0.04**	**98.15±0.04**
GraphSAGE	20%	90.17±0.60	88.62±0.67	88.09±0.74	85.62±0.84	94.26±0.14	94.44±0.16	89.58±2.68	87.45±3.28	86.04±1.19	84.25±1.55
VGAE		89.73±0.24	91.03±0.26	90.66±0.38	91.59±0.36	93.44±0.4	93.71±0.26	96.61±0.16	96.42±0.16	89.44±0.40	90.71±0.26
SEAL		91.40±0.22	92.80±0.13	89.13±0.23	91.33±0.06	93.22±0.16	93.90±0.13	94.68±0.09	95.26±0.14	90.13±0.23	91.46±0.27
S3GRL		92.18±0.32	92.01±0.58	94.76±0.56	94.42±0.68	95.55±0.52	95.16±0.71	95.32±0.27	95.17±0.94	92.85±0.58	93.94±0.77
S2GAE		93.00±0.37	92.42±0.60	94.82±0.22	94.74±0.26	96.60±0.15	96.24±0.15	97.76±0.09	97.19±0.13	97.21±0.10	96.65±0.13
MGAE		95.75±0.11	96.23±0.09	95.75±0.11	96.23±0.09	97.47±0.03	97.20±0.03	98.51±0.01	98.29±0.02	98.11±0.03	97.88±0.03
CGCL		**96.61±0.35**	**97.16±0.15**	**97.21±0.18**	**97.51±0.13**	**98.27±0.02**	**98.14±0.04**	**98.77±0.01**	**98.59±0.01**	**98.32±0.02**	**98.11±0.03**

Table 1: Link prediction results obtained on all datasets.

Methods	Testing ratios	Cora		CiteSeer		PubMed		Photo		Computers	
		AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
DGCL$_{\textbf{one-view}}$	10%	96.77±0.13	97.13±0.12	97.10±0.17	97.20±0.19	98.30±0.03	98.21±0.04	98.63±0.04	98.43±0.05	98.08±0.03	97.96±0.04
CGCL		**97.00±0.15**	**97.34±0.11**	**97.26±0.23**	**97.54±0.16**	**98.48±0.03**	**98.37±0.03**	**98.88±0.01**	**98.72±0.02**	**98.39±0.04**	**98.41±0.04**
DGCL$_{\textbf{one-view}}$	20%	96.91±0.13	96.95±0.12	97.09±0.12	97.18±0.14	98.19±0.03	98.02±0.04	98.61±0.05	98.45±0.04	98.01±0.03	97.91±0.05
CGCL		**97.61±0.35**	**97.16±0.15**	**97.28±0.18**	**97.51±0.13**	**98.27±0.02**	**98.14±0.04**	**98.77±0.01**	**98.59±0.01**	**98.32±0.02**	**98.37±0.03**

Table 2: Ablation study concerning the main training stages of the proposed CGCL method conducted on all datasets.

Each graph dataset is divided into three parts, including a training set, a validation set and a testing set. We use two different sets of percentages for the validation set and testing set, including (1) 5% of the validation set and 10% of the testing set and (2) 10% of the validation set and 20% of the testing set. The links in the validation and testing sets are masked in the training graph structure. For example, we randomly select 5% and 10% of the links and the same numbers of disconnected node pairs as testing and validation sets under the first setting, respectively. The remainder of the links in the graph structure are used for training.

Comparison Methods We compare the proposed CGCL method with several state-of-the-art methods for link prediction, including GraphSAGE (Hamilton, Ying, and Leskovec 2017), a VGAE (Veličkovi’c et al. 2018), SEAL (Zhang et al. 2021), S3GRL (Louis, Jacob, and Salehi-Abari 2023), S2GAE (Tan et al. 2023), and an MGAE (Li et al. 2022a). The source codes of the competing algorithms are provided by their respective authors. For the MGAE, the edgewise random masking strategy is chosen to sample a subset of the edges in each dataset.

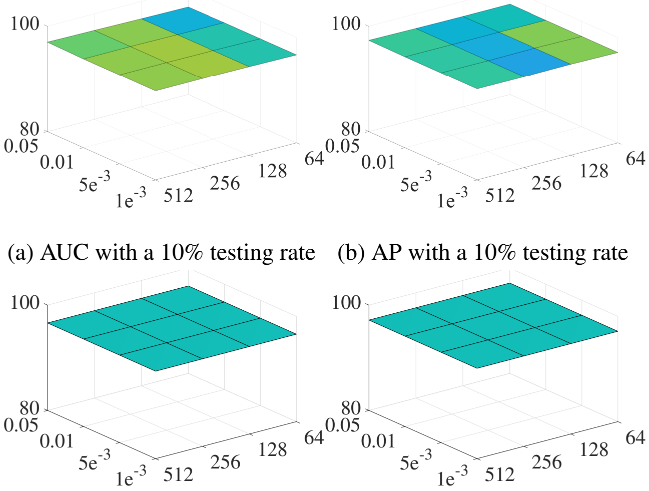
Evaluation Metrics Two metrics are utilized to evaluate the link prediction performance of all competing algorithms, including the area under the curve (AUC) score and average precision (AP) score. For comparison, each experiment is conducted 10 times with different random parameter initializations. We report the mean values and standard deviations achieved by all the competing methods on the five graph datasets. For each evaluation metric, a higher value represents better link prediction performance.

Parameter Settings The proposed network architecture contains 2 hidden layers in the CGCL model. The sizes of the 2 hidden layers are set to $[d_v, d_v/2]$, where d_v is the number of neural units in the first hidden layer. In the experiments, d_v ranges within $\{512, 256, 128, 64\}$. The learning rate of the proposed CGCL method r is chosen from $\left\{1e^{-3}, 5e^{-3}, 0.01, 0.05\right\}$. For all datasets, the number of iterations is set to 800 during the training stage. To conduct a fair comparison, the best link prediction results of these competing methods are obtained by tuning their parameters.

Performance Evaluation

The experimental results produced by all competing methods on the five link prediction tasks are reported in Table 1. The best and second-best values of the link prediction results are highlighted in bold and underlined, respectively. We observed that the proposed CGCL method almost performs better than the other competing methods in terms of the AUC and AP. For example, CGCL achieves performance improvements of approximately 0.26%, 0.86%, 0.24% and 0.13% in terms of the AUC with a testing rate of 10% on the Cora, PubMed, Photo and Computers datasets, respectively. Moreover, CGCL outperforms the competing methods in terms of the AUC and AP metrics as the testing rate increases from 10% to 20% in the link prediction tasks. These results demonstrate the superiority of CGCL over the other methods.

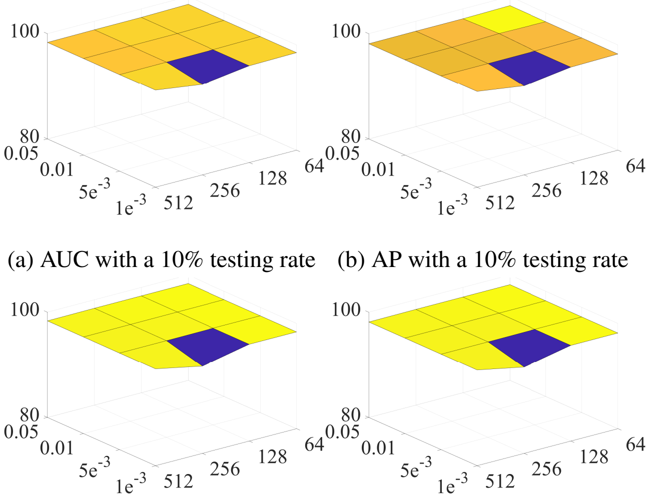
As expected, the AUC and AP results produced by CGCL slightly decrease as the testing rate increases from 10% to 20% on the three citation datasets, including the Cora, Citeseer and PubMed datasets. In contrast, the AUC and AP results of CGCL remain almost unchanged on the Photo and



(a) AUC with a 10% testing rate (b) AP with a 10% testing rate

(c) AUC with a 20% testing rate (d) AP with a 20% testing rate

Figure 2: The AUC and AP values yielded by CGCL with different combinations of d_v and r on the Cora dataset.



(a) AUC with a 10% testing rate (b) AP with a 10% testing rate

(c) AUC with a 20% testing rate (d) AP with a 20% testing rate

Figure 3: The AUC and AP values yielded by CGCL with different combinations of d_v and r on the Computers dataset.

Computers datasets. These two datasets contain larger numbers of edges than the other datasets. This demonstrates the effectiveness and robustness of the coupled graph structure augmentation approach.

Two main reasons highlight the advantages of the proposed CGCL method. First, constructing two complementary augmented views of a graph structure enhances the diversity of the produced graph representations while counteracting the adverse consequences of information losses in the graph representations. Additionally, the MGAE yields encouraging AUC and AP results in the experiments due to its effective edge masking strategy. Second, the proposed CGCL method achieves invariant graph representations through cross-view graph consistency learning. This mechanism plays a key role in facilitating the reconstruction of the incomplete graph structure within the scope of self-supervised learning.

Ablation Study

In Section , CGCL constructs two complementary augmented graph structure views using the coupled graph structure augmentation scheme. To verify the importance of the proposed graph structure augmentation scheme in CGCL, we further conduct an ablation study to isolate the necessity of the two complementary augmented views. Specifically, we consider a special version, i.e., a variant that only chooses one of the two complementary augmented views during the training stage, which is referred to as DGCL$_{\textbf{one-view}}$. We employ the same experimental settings as those utilized above. The best experimental results derived from the two complementary augmented views are included for comparison purposes. Table 2 shows the experimental AUC and AP results produced by DGCL$_{\textbf{one-view}}$ and CGCL. We see that CGCL performs better than DGCL$_{\textbf{one-view}}$ on the link prediction tasks. This provides strong empirical evidence demonstrating the importance of the two complementary augmented views in CGCL.

Parameter Sensitivity Study

We conduct experiments to investigate the sensitivity levels of the two parameters in the proposed CGCL method, including the number of neural units in the first hidden layer d_v and the learning rate r. The d_v and r parameters range within $\{512, 256, 128, 64\}$ and $\{1e^{-3}, 5e^{-3}, 0.01, 0.05\}$, respectively. Due to space limitations, two representative datasets are selected for evaluation, i.e., the Cora and Computers datasets. Figs. 2 and 3 show the experimental results achieved by CGCL in terms of the AUC and AP values obtained with different combinations of d_v and r. CGCL can achieve relatively stable link prediction results under different testing ratios with different combinations of d_v and r. This indicates that CGCL performs well across relatively large d_v and r ranges on the Cora and Computers datasets.

Conclusion

In this paper, we present a CGCL method that learns invariant graph representations from graph-structured data. CGCL utilizes a coupled graph structure augmentation scheme to construct two complementary augmented graph structure views. This augmentation scheme supports graph consistency learning, thereby enhancing the generalizability of the graph representations produced by CGCL. Through cross-view graph consistency learning, CGCL effectively acquires invariant graph representations and facilitates the construction of an incomplete graph structure. Extensive experimental results obtained on graph datasets demonstrate that the proposed CGCL method almost outperforms several state-of-the-art approaches.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grants U21B2040, 62176171 and 72374089, in part by the Fundamental Research Funds for the Central Universities under Grant CJ202303, and in part by Sichuan Science and Technology Planning Project under Grant 24NSFTD0130.

References

Chen, J.; Mao, H.; Peng, D.; Zhang, C.; and Peng, X. 2023a. Multiview clustering by consensus spectral rotation fusion. *IEEE Trans. Image Process.*, 5153–5166.

Chen, J.; Mao, H.; Woo, W. L.; ; and Peng, X. 2023b. Deep multiview clustering by contrasting cluster assignments. In *Proc. 19th IEEE Int. Conf. Comput. Vis.*, 16752–16761. Paris, France.

Chen, J.; Wang, Z.; Mao, H.; and Peng, X. 2023c. Low-rank tensor learning for incomplete multiview clustering. *IEEE Trans. Knowl. Data Eng.*, 11556–11569.

Cover, T. M.; and Thomas, J. A. 2006. Elements of Information Theory, 2nd Edition. In *Wiley*.

Cui, G.; Zhou, J.; Yang, C.; and Liu, Z. 2020. Adaptive graph encoder for attributed graph embedding. In *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 976–985.

Fang, R.; Wen, L.; Kang, Z.; and Liu, J. 2022. Structure-preserving graph representation learning. In *IEEE Int. Conf. Data Min.*, 927–932. Orlando, Florida, USA.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proc. 31th Adv. Neural Inf. Process. Syst.*, 1–11. Long Beach, CA, USA.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proc. 37th Int. Conf. Mach. Learn.*, 4116–4126.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; and Wang, C. 2022. GraphMAE: Self-supervised masked graph autoencoders. In *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 594–604.

Kipf, T. N.; and Welling, M. 2016. Variational graph autoencoders. *arXiv preprint*, 1–3.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. 5th Int. Conf. Learn. Represent.*, 1–14. Toulon, France.

Li, J.; Wu, R.; Sun, W.; L. Chen, S. T.; Zhu, L.; Meng, C.; Zheng, Z.; and Wang, W. 2022a. What's behind the mask: understanding masked graph modeling for graph autoencoders. *arXiv preprint*, 1–13.

Li, Y.; Yang, M.; Peng, D.; Li, T.; Huang, J.; and Peng, X. 2022b. Twin contrastive learning for online clustering. *Int. J. Comput. Vis.*, 130: 2205–2221.

Liu, X.; Wang, R.; Zhou, J.; Chen, C. L. P.; Zhang, T.; Chen, Y.; Han, S.; Du, T.; Ji, K.; and Zhang, K. 2023a. Transfer learning-based collaborative multiview clustering. *IEEE Trans. Fuzzy Syst.*, 31: 1163–1177.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. S. 2023b. Graph self-supervised learning: A survey. *IEEE Trans. Knowl. Data Eng.*, 35: 5879–5900.

Louis, P.; Jacob, S. A.; and Salehi-Abari, A. 2023. Simplifying subgraph representation learning for scalable link prediction. *arXiv preprint*, 1–9.

McAuley, J.; Targett, C.; Shi, Q.; and Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proc. 38th Int. ACM SIGIR Conf. R. D. Inf. Retr.*, 43–52.

Namata, G.; London, B.; Getoor, L.; and Huang, B. 2012. Query-driven active surveying for collective classification. In *10th Int. WS. Min. Learn. Graphs*, 43–52.

Peng, X.; Li, Y.; Tsang, I. W.; H. Zhu, J. L.; and Zhou, J. T. 2022. XAI beyond classification: interpretable neural clustering. *J. Mach. Learn. Res.*, 23(6): 1–28.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29: 93–93.

Tan, Q.; Liu, N.; Huang, X.; Choi, S.; Li, L.; Chen, R.; and Hu, X. 2023. S2GAE: self-supervised graph autoencoders are generalizable learners with graph masking. In *Proc. 16th Int. Conf. Web Search Data Min.*, 787–795. Singapore.

Tsai, Y.; Zhao, H.; Yamada, M.; Morency, L.; and Salakhutdinov, R. 2020. Neural methods for point-wise dependency estimation. In *Proc. 34th Adv. Neural Inf. Process. Syst.*, 62–72. Online.

Veličkovi'c, P.; Cucurull, G.; Casanova, A.; Romero, A.; Li.o, P.; and Bengio, Y. 2018. Graph attention networks. In *Proc. 6th Int. Conf. Learn. Represent.*, 1–12. Vancouver, BC, Canada.

Wang, H.; Guo, X.; Deng, Z.; and Lu, Y. 2022. Rethinking minimal sufficient representation in contrastive learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 16041–16050. New Orleans, Louisiana, USA.

Wang, Y.; Han, S.; Zhou, J.; Chen, L.; Chen, C. P.; Zhang, T.; Liu, Z.; Wang, L.; and Chen, Y. 2023. Random feature based collaborative kernel fuzzy clustering for distributed peer-to-peer networks. *IEEE Trans. Fuzzy Syst.*, 31: 692–706.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proc. 44th Int. ACM SIGIR Conf. R. D. Inf. Retr.*, 726–735.

You, Y.; Chen, T.; Y, Y. S.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. In *Proc. 34th Adv. Neural Inf. Process. Syst.*, 5812–5823. Online.

Zhang, M.; Li, P.; Xia, Y.; Wang, K.; and Jin, L. 2021. Labeling trick: A theory of using graph neural networks for multi-node representation learning. In *Proc. 35th Adv. Neural Inf. Process. Syst.*, 9061–9073. Online.

Zhu, Y.; Xu, Y.; adn Q. Liu, F. Y.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint*, 1–9.