CSGCL: Community-Strength-Enhanced Graph Contrastive Learning

Han Chen[1,2], **Ziwen Zhao**[1], **Yuhua Li**[1,*], **Yixiong Zou**[1], **Ruixuan Li**[1] and **Rui Zhang**[3,*,†]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology
[2]Institute of Artificial Intelligence, Huazhong University of Science and Technology
[3]Tsinghua University
{HanChenHUST, zwzhao, idcliyuhua, yixiongz, rxli}@hust.edu.cn, rayteam@yeah.net

Abstract

Graph Contrastive Learning (GCL) is an effective way to learn generalized graph representations in a self-supervised manner, and has grown rapidly in recent years. However, the underlying community semantics has not been well explored by most previous GCL methods. Research that attempts to leverage communities in GCL regards them as having the same influence on the graph, leading to extra representation biases. To tackle this issue, we define "community strength" to measure the difference of influence among communities. Under this premise, we propose a **C**ommunity-**S**trength-enhanced **G**raph **C**ontrastive **L**earning (CSGCL) framework to preserve community strength throughout the learning process. Firstly, we present two novel graph augmentation methods, Communal Attribute Voting (CAV) and Communal Edge Dropping (CED), where the perturbations of node attributes and edges are guided by community strength. Secondly, we propose a dynamic "Team-up" contrastive learning scheme, where community strength is used to progressively fine-tune the contrastive objective. We report extensive experiment results on three downstream tasks: node classification, node clustering, and link prediction. CSGCL achieves state-of-the-art performance compared with other GCL methods, validating that community strength brings effectiveness and generality to graph representations. Our code is available at https://github.com/HanChen-HUST/CSGCL.

1 Introduction

Graph Representation Learning (GRL) is extensively used in knowledge graphs [Zhang *et al.*, 2022], recommendation [He *et al.*, 2020], e-commerce [Li *et al.*, 2020], biological systems [Rao *et al.*, 2022], etc. Unlike the label-dependent and noise-sensitive supervised GRL methods, Graph Contrastive Learning (GCL) [Becker and Hinton, 1992] extracts well-generalized representations between paired data views in a self-supervised manner. Many promising results from node-level GCL studies have been achieved [Veličković *et al.*, 2019; Hassani and Khasahmadi, 2020; Zhu *et al.*, 2021]. Despite their success, most of these studies hardly consider the community in GCL.

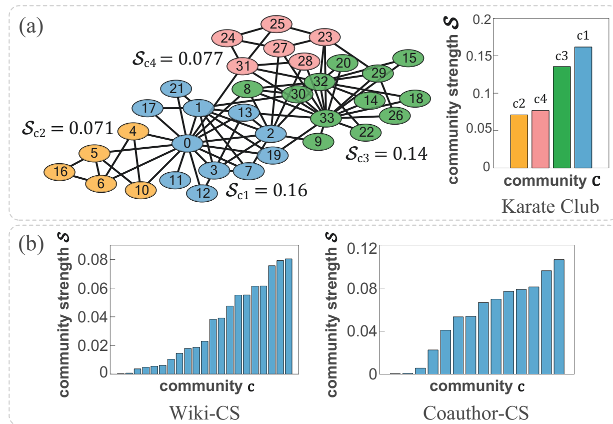


Figure 1: Community strength varies in real-world networks. (a) Community strength on the Karate Club dataset [Zachary, 1977]. Communities, marked in different colors, are subgraphs with dense internal links. (b) Community strength distributions of two real-world network examples: Wiki-CS [Zachary, 1977] and Coauthor-CS [Shchur *et al.*, 2018], which suggest that there are differences in community strength in real-world networks.

A **community** is a high-order structure of a graph, characterized by a denser connection among its members [Girvan and Newman, 2002]. This underlying structure is highly informative: for citation networks, communities indicate different academic fields; for co-purchase networks, communities indicate the common interests of a customer in multiple types of merchandise. Therefore, it is necessary to take into account communities in GCL. Recently, research like gCooL [Li *et al.*, 2022] combines community detection with GCL. However, their work is built on an underlying hypothesis that different communities have the same influence on global semantics, which may lead to extra representation biases in GCL.

To tackle this issue, we hold that **communities influence global semantics differently**, which is vital for GCL. We define "**community strength**" to measure this influence of

*Corresponding authors.
†Tsinghua University (www.ruizhang.info)

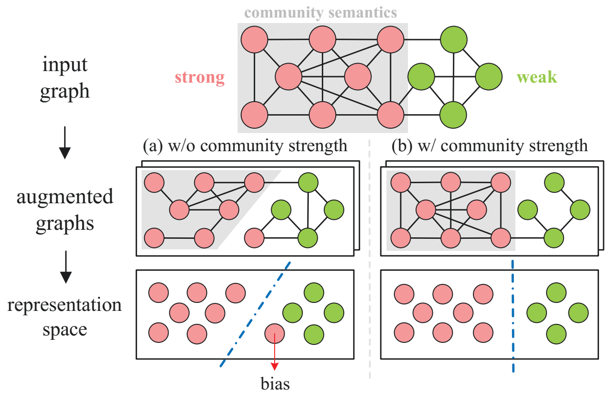


Figure 2: Illustration of the advantage of community strength to GCL. (a) Node representations without the guidance of community strength, where strong and weak communities are treated equally. Therefore, community semantics is changed during the augmentation, resulting in a bias in the representation space. (b) Node representations by CSGCL, where the bias is handled by preserving community strength information. Note that here we use a schematic subgraph example from some dataset with more communities.

communities, and give a visualized example of community strength in Figure 1(a). Community strength, denoted as $\mathcal{S}$, indicates **how dense the connection of a community c is**, and reflects the influence of c on the entire graph. A stricter definition can be found in Section 3.2. "Strong communities", with more densely connected edges, have a larger $\mathcal{S}$ (*e.g.* $\mathcal{S}_{c1}$ and $\mathcal{S}_{c3}$); "weak communities", more sparsely connected ones, have a smaller $\mathcal{S}$ (*e.g.* $\mathcal{S}_{c2}$ and $\mathcal{S}_{c4}$). As shown in Figure 1(b), the difference of community strength is one of the inherent features of real-world networks. Hence, community strength is of practical importance. We naturally ask: *is there an effective way to leverage community strength information in GCL?*

To answer this question, we propose a novel Community-Strength-enhanced Graph Contrastive Learning (CSGCL) framework. It can capture and preserve community strength information throughout the learning process, from data augmentation to the training objective. Firstly, we put forward two novel graph augmentation approaches, Communal Attribute Voting (CAV) and Communal Edge Dropping (CED), which apply community strength respectively to every attribute and every edge to preserve the differences among communities. Secondly, we propose a dynamic "Team-up" contrastive scheme for CSGCL to progressively fine-tune the contrastive objective with community strength.

With the help of community strength, CSGCL can learn more discriminative representations, as in Figure 2. Without the consideration of community strength, every community is treated equally, which is not the case for real-world networks, so it leads to more representation biases. CSGCL takes into account community strength to handle these biases.

Our main contributions are as follows:

- We give a definition of community strength to evaluate the influence of a community. To the best of our knowledge, we are the first to manage to preserve community strength information in GCL.
- We propose two enhanced graph augmentation methods, CAV and CED, in which the perturbations of node attributes and edges are guided by community strength.
- A dynamic Team-up objective is devised, which directs the training process to preserve community strength.
- Extensive experiments on graph benchmark datasets show that CSGCL achieves state-of-the-art performance on up to three downstream tasks – node classification, node clustering, and link prediction.

2 Related Work

In this section, we give a brief review of traditional GRL methods, node-level GCL methods, and GCL with communities. Then, we compare CSGCL with these methods. For additional related work, see Appendix D.

2.1 Graph Representation Learning (GRL)

GRL extracts and compresses the semantic information and topological structure of a graph into low-dimensional representation vectors. Traditional methods [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] employ random walks to generate node sequences and embed them into the representation space using text encoding approaches. These methods only consider the co-occurrence of nodes. As improvements, CPNE [Wang *et al.*, 2017] preserves community information by Non-negative Matrix Decomposition and, WGCN [Zhao *et al.*, 2021] captures weighted structure features. An effective GRL paradigm is graph self-supervised learning [Kipf and Welling, 2016; Liu *et al.*, 2022], which creates pretext tasks to leverage the information within data, helping the model to get rid of label dependency. However, they do not preserve community strength well in their representations.

2.2 Graph Contrastive Learning (GCL)

Shortly after the proposal of contrastive learning [Hjelm *et al.*, 2019; Chen *et al.*, 2020], it was introduced into the GRL field and became a new self-supervised hotspot. Most existing GCL methods [Zhu *et al.*, 2020; Lee *et al.*, 2022; Zhu *et al.*, 2022; Wei *et al.*, 2022] only focus on node-level relations. GraphCL [You *et al.*, 2020] introduces augmentations of nodes or node pairs like DropEdge [Rong *et al.*, 2020] and feature masking into GCL to increase the difference between two graph views. GCA [Zhu *et al.*, 2021] further improves graph augmentation methods to adaptively capture the impact of different nodes. Outstanding as they are, they **ignore the inherent community information in networks**. Other works which consider the global information of graphs [Veličković *et al.*, 2019; Hassani and Khasahmadi, 2020] also pay little attention to community semantics, which is discarded or undermined in node-level readout operations.

gCooL [Li *et al.*, 2022] makes a good attempt to combine community information with GCL. It partitions communities on the representation graph and contrasts node-level and community-level representations across two different views, considering node pairs of the same community as positive samples. However, **gCooL does not capture the difference between strong and weak communities**. The differences between CSGCL and the aforementioned methods are that (1)

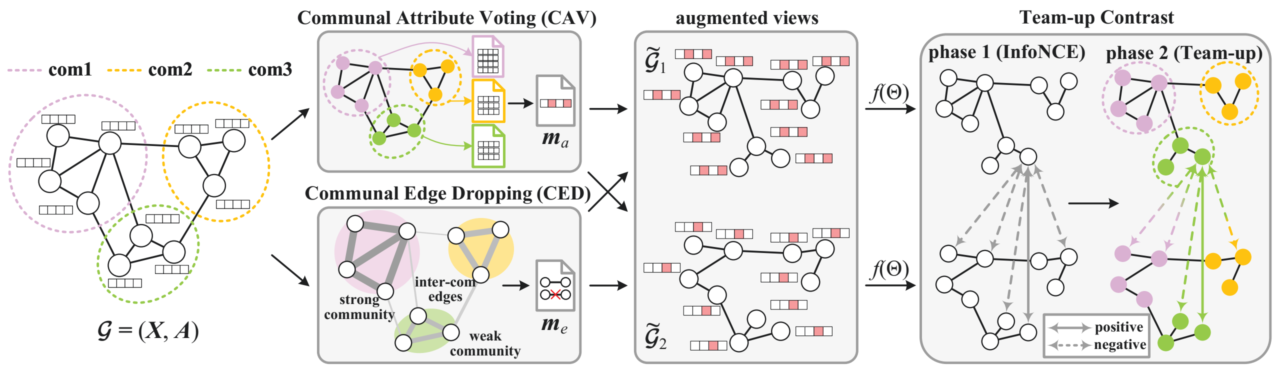


Figure 3: Overview of the proposed CSGCL framework. Communities of the input graph $\mathcal{G}$ are marked in three different colors. Firstly, two augmented views are generated from $\mathcal{G}$ using two concurrent graph augmentations, CAV and CED: CAV tends to remove the most-voted-for attributes, where each community submits a "vote" (colored file icons) with attribute penalties; CED tends to retain edges inside communities, especially the strong ones (thick), and drop inter-community edges (thin). Then, two augmented views are fed into a shared encoder and projector $f(\Theta)$ to get the representation graphs. In the end, the Team-up contrastive scheme (phase 2) progressively fine-tunes the InfoNCE objective (phase 1).

Symbol	Description
$\mathcal{G}, \tilde{\mathcal{G}}$	original & perturbed attributed graph
$\mathcal{V}$	node set of graph $\mathcal{G}$
$\mathcal{E}$	edge set of graph $\mathcal{G}$
$\mathbf{X}, \tilde{\mathbf{X}}$	original & perturbed node attribute matrix
$\mathbf{A}, \tilde{\mathbf{A}}$	original & perturbed adjacency matrix
$\mathbf{Y}$	classification labels of graph $\mathcal{G}$
$\mathbf{Z}$	representation matrix of graph $\mathcal{G}$
$\mathcal{C}$	community set of graph $\mathcal{G}$
$\mathcal{E}_c$	edge set of a community c
$\mathcal{S}$	vector of strength $\mathcal{S}_c$ of every community c
$\mathbf{H}$	community indicator matrix
$\mathbf{M}_{i:}$	ith row of any matrix $\mathbf{M}$
$\mathbf{M}_{:j}$	jth column of any matrix $\mathbf{M}$
$M_{i,j}$	element at row i and column j of any matrix $\mathbf{M}$
$\mathbf{M}^{(k)}$	belonging to the kth view of any matrix $\mathbf{M}$, $k = 1, 2$
$d(\cdot)$	degree of a node
$f(\cdot;\Theta)$	graph encoder with parameters Θ
$\mathcal{T}(\cdot)$	data augmentations
$\mathbb{1}_{[ex]}$	indicator function, $\begin{cases} 1, ex \text{ is true} \\ 0, ex \text{ is false} \end{cases}$
$\lvert\cdot\rvert$	size of a set
$\lVert\cdot\rVert$	Euclid norm
$\circ$	Hadamard product

Table 1: Notations.

CSGCL is a contrastive method taking advantage of community strength information; (2) CSGCL preserves community strength from data augmentation to model optimization.

3 Community-Strength-enhanced Graph Contrastive Learning

In this section, we illustrate and formulate the details of CSGCL. The overall architecture of CSGCL is shown in Figure 3. Notations used below are summarized in Table 1. For an intuitive workflow of CSGCL, we summarize our approaches in pseudo-codes in Appendix A.

3.1 Graph Contrastive Learning

An undirected attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$. In a graph dataset, a graph with n nodes and d attributes is given as a node attribute matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and a symmetric adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, denoted as $\mathcal{G} = (\mathbf{X}, \mathbf{A})$. For nodes $u, v \in \mathcal{V}$, $A_{(u,v)} = 1$ iff edge $(u, v) \in \mathcal{E}$, otherwise $A_{(u,v)} = 0$.

Graph Contrastive Learning aims to maximize the agreement of pairwise graph embeddings [Hjelm *et al.*, 2019]. In general, A GCL framework is comprised of a GNN encoder, a projection head, and a contrastive loss. At the beginning of training, GCL generates two graph views $(\tilde{\mathcal{G}}_1, \tilde{\mathcal{G}}_2)$ by random perturbation to the input graph. Such perturbation is usually from multiple data augmentations $\mathcal{T}(\mathcal{G})$, *e.g.* DropEdge [Rong *et al.*, 2020] and feature masking [Zhu *et al.*, 2020]. Then, two views are both fed into a shared GNN encoder (like a two-layer GCN [Kipf and Welling, 2017]) to obtain node representations (but only the encoder is retained for various downstream predictions). Let $\mathbf{Z} = f(\tilde{\mathcal{G}}; \Theta)$ be the projected embedding of $\tilde{\mathcal{G}}$, where f is the graph encoder followed by a nonlinear projection layer. The embeddings of a certain node i in different views $(\mathbf{Z}^{(1)}_{i:}, \mathbf{Z}^{(2)}_{i:})$ are seen as positive pairs, and all other pairs (embeddings of different nodes in the same or different views) are seen as negative pairs.

GCL usually employs noise contrastive estimation (NCE) as the optimization objective. The most commonly used one is InfoNCE [Oord *et al.*, 2018], which enhances the discrimination of representations by pulling similar (positive) ones together and pushing dissimilar (negative) ones away. The similarity between node pairs is calculated as

$$s^{(1,2)}_{ij} = sim(\mathbf{Z}^{(1)}_{i:}, \mathbf{Z}^{(2)}_{j:})/\tau, \tag{1}$$

where $sim(\mathbf{z}_i, \mathbf{z}_j) = (\mathbf{z}_i^\top \mathbf{z}_j)/\left(\|\mathbf{z}_i\|\|\mathbf{z}_j\|\right)$, and τ is the temperature of the similarity. Then InfoNCE is formulated as

$$\mathcal{L} = \mathbb{E}_{(\tilde{\mathcal{G}}_1, \tilde{\mathcal{G}}_2) \sim \mathcal{T}(\mathcal{G})} \left(-\frac{1}{n} \sum_{i=1}^{n} \log \frac{\exp(s^{(1,2)}_{ii})}{\sum_{\substack{j=1 \\ j \neq i}}^{n} \exp(s^{(1,1)}_{ij}) + \sum_{j=1}^{n} \exp(s^{(1,2)}_{ij})} \right). \tag{2}$$

3.2 Community-strength-enhanced Augmentations

In this section, we introduce the community-strength-enhanced augmentations: Communal Attribute Voting and Communal Edge Dropping.

Community strength. To begin with, communities in the graph are highlighted by unsupervised community detection methods. CSGCL utilizes existing community detectors so that our study can refocus attention on the communities in graph representations. We give more details of community detection methods suitable for CSGCL in Appendix D.

Let $\mathcal{C}$ be the set of all communities in graph $\mathcal{G}$. Then, we have the definition below:

Definition 1. (Community strength $\mathcal{S}$) For every community c of an undirected attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $|\mathcal{E}|$ and $|\mathcal{E}_c|$ be the number of edges of the entire graph and inside that community respectively. Then, the community strength $\mathcal{S}_c$ of community c is formulated as follows:

$$\mathcal{S}_c = \frac{|\mathcal{E}_c|}{|\mathcal{E}|} - \frac{(\sum_{v \in c} d(v))^2}{4|\mathcal{E}|^2} \tag{3}$$

which is composed of two terms: the proportion of edges inside community c, and the inter-community connection between c and other communities.

Community strength is based on modularity [Newman and Girvan, 2004], a measure of graph partition quality, which compares every community with its randomly connected counterpart, and calculates the difference in their number of edges. However, community strength is a metric of the local topological structure, while modularity is a quality metric of the entire graph. $\mathcal{S} > 0$ for every community, because $\mathcal{S}_c \leq 0$ means c does not have any community characteristics.

Communal Attribute Voting (CAV). Based on the definition above, we propose CAV, a graph augmentation method based on attribute masking, which adopts community voting for attribute removal: **attributes which are more influential in strong communities are better preserved, whereas less influential attributes are more likely to be voted out.**

As the voting begins, a community penalty w_a is assigned to every single attribute of $\mathbf{X}$. Intuitively, w_a reflects the odds to be removed for every attribute candidate:

$$\mathbf{w}_a = \tilde{n}_a \left(\log(\text{abs}(\mathbf{X})^\top \mathbf{H}\mathcal{S})\right) \tag{4}$$

where $\mathbf{H} \in \{0,1\}^{n \times |\mathcal{C}|}$ is an indicator matrix with $H_{i,c} = \mathbb{1}_{[v_i \in c]}$ indicating to which community the ith node belongs, and $\tilde{n}_a(x) = (x_{max} - x)/(x_{max} - x_{mean})$ is a one-dimensional normalization operation. Each node will score and vote for its redundant attributes, and the least-voted-for attributes will win a greater opportunity to be preserved.

Then, inspired by [Zhu *et al.*, 2021], we adaptively adjust the attribute perturbation distribution using $\mathbf{w}_a$. Specifically, the voting results are a group of attribute masks $\mathbf{m}_a$ independently sampled from two Bernoulli distributions:

$$\mathbf{m}_a^{(1)} \sim Bernoulli\left(1 - \mathbf{w}_a p_a^{(1)}\right), \; \mathbf{m}_a^{(2)} \sim Bernoulli\left(1 - \mathbf{w}_a p_a^{(2)}\right) \tag{5}$$

where $p_a^{(1)}$ and $p_a^{(2)}$ are two hyperparameters controlling the sampling ranges. In this way, community strength is well preserved in the perturbed attribute matrices:

$$\tilde{\mathbf{X}}^{(1)} = \mathbf{m}_a^{(1)} \circ \mathbf{X}, \; \tilde{\mathbf{X}}^{(2)} = \mathbf{m}_a^{(2)} \circ \mathbf{X}, \tag{6}$$

where $\circ$ is the Hadamard product.

Communal Edge Dropping (CED). The edge is the fundamental structural unit of graphs. Evolved from DropEdge [Rong *et al.*, 2020], CED preserves community structures from perturbations guided by community strength.

The rationale for CED is that (1) **intra-community edges are more important than inter-community edges**, and (2) **edges in strong communities are more important than those in weak communities**. If there is a scoring function $w(e)$ to calculate the weight w_e of each edge $e \in \mathcal{E}_c$ in the community c, it must meet the following condition:

$$w_e = w(e), \; s.t. \; w(e_{strong}) > w(e_{weak}) > w(e_{inter}) \tag{7}$$

where e_{strong} is an intra-strong-community edge, e_{weak} is an intra-weak-community edge with $\mathcal{S}_{strong} > \mathcal{S}_{weak}$, and e_{inter} is an inter-community edge.

To this end, let $e = (u_i, u_j)$, we formulate CED as:

$$w_e = \tilde{n}_e \left(w(e)\right), \tag{8}$$

$$w(e) = \begin{cases} \mathbf{H}_{i:}\mathcal{S}, & (\mathbf{H}\mathbf{H}^\top \circ \mathbf{A})_{i,j} = 1 \\ -\left(\mathbf{H}_{i:}\mathcal{S} + \mathbf{H}_{j:}\mathcal{S}\right), & \text{otherwise} \end{cases} \tag{9}$$

where $\tilde{n}_e(x) = (x - x_{min})/(x_{mean} - x_{min})$ is a one-dimensional normalization operation.

Theorem 1. $\tilde{n}_e(w(e))$ *defined in* (8)-(9) *satisfies* (7).

Proof. See Appendix B. □

Remark 1. *Theorem 1 indicates that CED is an ideal graph augmentation on edges: (1) strong communities have larger edge weights and vice versa; (2) inter-community edges have the smallest weights, which are much more likely to be dropped. Such design aims to retain real-world community properties in the perturbed graphs as much as possible.*

Then, similar to CAV, the edge weight vector $\mathbf{w}_e$ is used to adjust the edge perturbation distribution:

$$\mathbf{m}_e^{(1)} \sim Bernoulli\left(\mathbf{w}_e p_e^{(1)}\right), \mathbf{m}_e^{(2)} \sim Bernoulli\left(\mathbf{w}_e p_e^{(2)}\right) \tag{10}$$

where $p_e^{(1)}$ and $p_e^{(2)}$ function like $p_a^{(1)}$ and $p_a^{(2)}$. In this way, community strength is well preserved in the perturbed adjacency matrices:

$$\tilde{\mathbf{A}}^{(1)} = \left[m_{e,(u,v)}^{(1)} A_{(u,v)}^{(1)}\right], \; \tilde{\mathbf{A}}^{(2)} = \left[m_{e,(u,v)}^{(2)} A_{(u,v)}^{(2)}\right] \tag{11}$$

Up to now, two augmented graphs $(\tilde{\mathcal{G}}^{(1)}, \tilde{\mathcal{G}}^{(2)})$ have been generated by CSGCL to get embeddings.

3.3 Team-up Contrastive Learning Scheme

After obtaining the embeddings of two augmented graph views $\mathbf{Z}^{(1)} = f(\tilde{\mathcal{G}}^{(1)}; \Theta), \mathbf{Z}^{(2)} = f(\tilde{\mathcal{G}}^{(2)}; \Theta)$, we find it necessary to teach our encoder the differences among communities, guided by a community-strength-enhanced objective.

GRL with communities is analogous to real-world social group activities. On such occasions, people tend to team up with those with whom they are familiar. A group of strangers need to communicate with one another to seek common interests before they coalesce [Khambatti *et al.*, 2002]. Going back to the GRL scenario: at the beginning of training, nodes are unfamiliar with each other and crave for interaction with anyone within reach. As the training goes on, sufficient information exchange between nodes makes them understand their partners better. Driven by common interests, they tend to gather into groups – at this moment, the guidance of communities is indispensable for model optimization.

Inspired by this, we propose the **dynamic Team-up contrastive learning scheme** to learn communities more effectively. We divide CSGCL's learning process into two phases:

Individual contrast. In the first phase, the model optimizes InfoNCE to learn initial similarities between each node individuals. The form of InfoNCE is shown in (2).

Team-up contrast. In the second phase, every similarity term s of InfoNCE is fine-tuned by community strength:

$$\mathcal{L} = \mathbb{E}_{(\tilde{\mathbf{Z}}^{(1)},\tilde{\mathbf{Z}}^{(2)})} \left(-\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp(\tilde{s}_{ii}^{(1,2)})}{\sum_{\substack{j=1\\ j\neq i}}^{n} \exp(\tilde{s}_{ij}^{(1,1)}) + \sum_{j=1}^{n} \exp(\tilde{s}_{ij}^{(1,2)})} \right) \tag{12}$$

where

$$\tilde{s}_{ij}^{(1,2)} = s_{ij}^{(1,2)} + \gamma(\mathbf{H}_{i:} + \mathbf{H}_{j:})\mathcal{S}. \tag{13}$$

and $\mathbf{H}_{i:}$ is the community membership for the ith node. The latter term in (13) refers to the strength of "teams" (communities) to which the ith and jth nodes belong, and γ is a coefficient of community strength.

The final loss is a piecewise function combining the two phases above, with a dynamic change of the fine-tuned similarity $\tilde{s}$ in (13):

$$\tilde{s}_{ij}^{(1,2)} = s_{ij}^{(1,2)} + \gamma(t)(\mathbf{H}_{i:} + \mathbf{H}_{j:})\mathcal{S} \tag{14}$$

in which $\gamma(t)$ is a monotonically non-decreasing function that varies with training time t (in the units of 100 epochs). We simply consider a hard Sigmoid-shaped form for $\gamma(t)$:

$$\gamma(t; t_0, \gamma_{max}) = \min\{\max\{0, t - t_0\}, \gamma_{max}\} \tag{15}$$

where t_0 is the demarcation point of two phases. Thus we can unify two phases and formulate the final loss as (12)(14)(15): during the individual contrast ($t \leq t_0$, $\gamma = 0$), free communications are allowed between node individuals; during the Team-up contrast ($t_0 < t < t_0 + \gamma_{max}$) when the node-level relationships are well-learned, γ rises gradually to direct the model to notice community strength; when teaming-up is complete ($t \geq t_0 + \gamma_{max}$), we set $\gamma \equiv \gamma_{max}$ to prevent deviation from the contrastive objective.

4 Experiments

In this section, we describe our experiments that were conducted to evaluate our model and answer the questions below:

- Does GCL really benefit from our proposed methods? (Section 4.2).
- Is the boost to performance really given by community strength? (Section 4.3).
- How does the Team-up strength coefficient influence the performance on a certain graph? (Section 4.4).

Detailed experiment configurations can be found in Appendix C. Additional experiment results using different community detectors and metrics (micro- & macro-F1 and average precision) can be found in Appendix E.

4.1 Experiment Setup

Datasets. We use four benchmark graphs in different fields, including one directed graph: Wiki-CS; and three undirected graphs: Amazon-Computers (Computers), Amazon-Photo (Photo), and Coauthor-CS. We convert Wiki-CS to an undirected graph only during the community detection process by adding reverse edges. There is no other difference between Wiki-CS and undirected graphs for model training.

Baselines. We divide all baseline models into the following three categories:

- Traditional unsupervised models: Logistic regression (LogReg) & K-means [MacQueen, 1967] with raw features, DeepWalk [Perozzi *et al.*, 2014] w/ or w/o the use of node attributes, node2vec [Grover and Leskovec, 2016], and CPNE [Wang *et al.*, 2017];
- Supervised graph neural network models: GCN [Kipf and Welling, 2017] and GAT [Veličković *et al.*, 2018];
- Renowned and up-to-date self-supervised GRL models: GAE & VGAE [Kipf and Welling, 2016], DGI [Veličković *et al.*, 2019], MVGRL [Hassani and Khasahmadi, 2020], GRACE [Zhu *et al.*, 2020], GCA [Zhu *et al.*, 2021], AFGRL [Lee *et al.*, 2022], and gCooL [Li *et al.*, 2022].

For GCA and gCooL which have multiple model configurations, we select the best one for each dataset, marked as "-best" in the statistics.

Evaluation protocol. For node classification, we follow the evaluation protocol of [Zhu *et al.*, 2021] which trains and tests an ℓ_2-regularized logistic regression classifier with 10 random data splits (20 fixed splits for Wiki-CS). For node clustering, a K-means model [MacQueen, 1967] with fixed initial clusters is fit. For link prediction, the cosine similarity between every two nodes is calculated to evaluate the existence of a link. Each experiment is repeated 10 times to report the average performance along with the standard deviation.

Metrics. We use accuracy for node classification, normalized mutual information (NMI) [Ana and Jain, 2003] for node clustering, and area under the curve (AUC) [Zhou *et al.*, 2009] for link prediction.

Implementation details. We choose Leiden [Traag *et al.*, 2019] as the community detector for CSGCL, before which we pretested the community detection methods detailed in Appendix E.1. To ensure fairness, A simple two-layer GCN is employed to CSGCL as well as all other contrastive baselines. We use the Adam optimizer to optimize the model. Detailed environment configurations can be found in Appendix C.2.

Method	Training data	Level	Wiki-CS	Computers	Photo	Coauthor-CS
Raw (LogReg)	$\mathbf{X}$	–	71.85±0.00	73.25±0.00	79.02±0.00	89.64±0.00
DeepWalk (w/o $\mathbf{X}$)	$\mathbf{A}$	node	73.84±0.16	85.77±0.58	89.06±0.43	84.71±0.35
node2vec	$\mathbf{A}$	node	75.52±0.17	86.19±0.26	88.86±0.43	86.27±0.22
DeepWalk (w/ $\mathbf{X}$)	$\mathbf{X}, \mathbf{A}$	node	77.21±0.03	86.28±0.07	90.05±0.08	87.70±0.04
CPNE	$\mathbf{A}$	community	65.53±0.58	73.66±0.83	82.39±1.23	OOM
GAE	$\mathbf{X}, \mathbf{A}$	node	70.15±0.01	85.27±0.19	91.62±0.13	90.01±0.71
VGAE	$\mathbf{X}, \mathbf{A}$	node	75.63±0.19	86.37±0.21	92.20±0.11	92.11±0.09
DGI	$\mathbf{X}, \mathbf{A}$	node	75.35±0.14	83.95±0.47	91.61±0.22	92.15±0.63
MVGRL	$\mathbf{X}, \mathbf{A}$	node	77.52±0.08	87.52±0.11	91.74±0.07	92.11±0.12
GRACE	$\mathbf{X}, \mathbf{A}$	node	77.68±0.34	88.29±0.11	92.52±0.34	92.50±0.08
GCA-best	$\mathbf{X}, \mathbf{A}$	node	78.20±0.04	87.99±0.13	92.06±0.27	92.81±0.19
AFGRL	$\mathbf{X}, \mathbf{A}$	node	77.62±0.49	89.88±0.33	93.22±0.28	93.27±0.17
gCooL-best	$\mathbf{X}, \mathbf{A}$	community	78.20±0.09	88.67±0.10	92.84±0.20	92.75±0.01
CSGCL (ours)	$\mathbf{X}, \mathbf{A}$	community	**78.60±0.13**	**90.17±0.17**	**93.32±0.21**	**93.55±0.12**
GCN	$\mathbf{X}, \mathbf{A}, \mathbf{Y}$	–	78.02±0.51	87.79±0.36	91.82±0.01	93.06±0.00
GAT	$\mathbf{X}, \mathbf{A}, \mathbf{Y}$	–	77.62±0.69	88.64±0.63	92.16±0.47	91.49±0.30

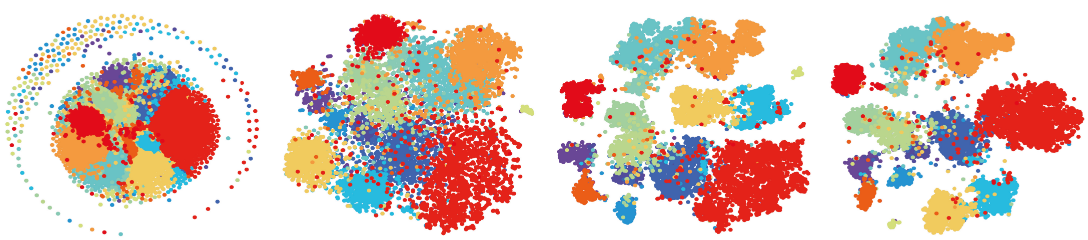
Table 2: Node classification results measured by accuracy (%). "OOM" stands for Out-Of-Memory on an 11GB GPU.

Method	Wiki-CS	Computers	Photo	Coauthor-CS
Raw (K-means)	18.22±0.00	16.59±0.00	28.22±0.00	64.18±0.00
CPNE	32.44±1.30	42.51±1.60	53.62±1.58	OOM
DGI	31.00±0.02	31.80±0.02	37.60±0.03	74.70±0.01
MVGRL	26.30±1.00	24.40±0.00	34.40±4.00	74.00±1.00
GRACE	28.68±1.18	38.97±0.91	47.70±4.28	73.79±0.58
GCA-best	30.22±1.09	39.09±0.55	51.37±4.15	74.38±0.42
gCooL-best	30.92±1.12	42.70±0.96	58.50±2.98	71.42±1.16
CSGCL (ours)	**32.80±1.50**	**45.09±0.95**	**58.79±2.17**	**78.29±1.24**

Table 3: Node clustering results measured by NMI (%).

Method	Computers	Photo	Coauthor-CS
Raw	54.58±0.00	60.21±0.00	93.69±0.00
GRACE	89.97±0.25	88.64±1.17	87.67±0.10
GCA-best	90.67±0.30	89.61±1.46	88.05±0.00
gCooL-best	90.45±0.86	90.83±2.05	88.91±0.04
CSGCL (ours)	**94.95±1.70**	**91.63±1.37**	**96.45±0.15**
GCN	87.89±0.90	88.20±0.08	92.71±0.63
GAT	87.60±2.23	89.32±2.06	92.80±0.75

Table 4: Link prediction results measured by AUC (%).



(a) Raw (b) node2vec (c) gCooL (d) CSGCL

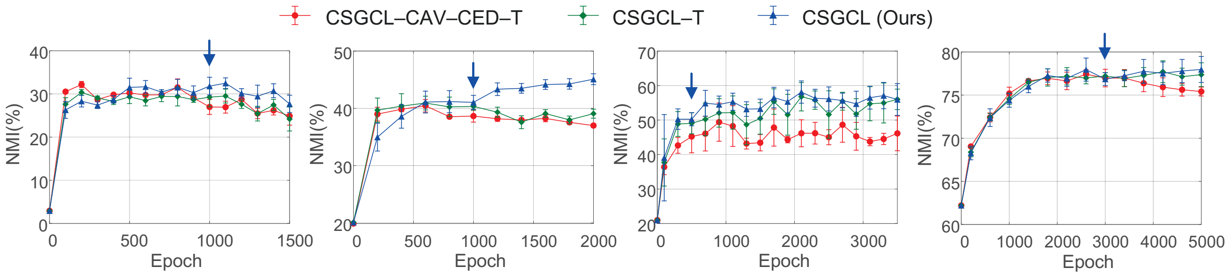
Figure 4: t-SNE visualization of representations on Coauthor-CS.

4.2 Overall Performance

In this section, we discuss the overall performance of CSGCL. CSGCL is adapted to three downstream tasks: node classification and node clustering on all datasets (Table 2 and 3), and link prediction on three undirected datasets (Table 4). Some statistics are borrowed from either their original papers or [Zhu *et al.*, 2021; Li *et al.*, 2022]. Data components used by each method during training are shown in the "Training data" column of Table 2, including node attributes $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, and labels $\mathbf{Y}$. The representation level of each unsupervised method is listed in the "Level" column, including node-level and community-level. **Bolded** results in Tables 2–5 below represent the best results for each column.

We can see that CSGCL is superior to not only the traditional baselines but also the latest GCL models in all three downstream tasks. This is most noticeable in node clustering, where the NMI score of CSGCL is 1.7% higher on average than the next-best methods; for link prediction, CSGCL has a 7.5% increase of AUC on Coauthor-CS compared with other contrastive methods. CSGCL is also observed to be competitive against fully supervised models, especially on link prediction. Such achievements reveal the great generality of graph representations of CSGCL: **community strength is beneficial to multiple downstream tasks on graphs.**

Note that gCooL – another GCL method with communities – also achieves great performance on these tasks, which backs up the benefit of community semantics for GCL. However, gCooL does not capture the differences among communities. We conduct the one-tailed Wilcoxon signed-rank test [Wilcoxon, 1992] to further back up our improvement: taking node classification as an example, we have $p = 9.77e\text{-}4 < 0.05$ on all datasets, indicating the acceptance of alternative hypothesis that CSGCL has significant improvements over the best model of gCooL.



(a) Wiki-CS (b) Computers (c) Photo (d) Coauthor-CS

Figure 5: NMI (%) of community detection in the ablation configurations. "↓" points out t_0, the demarcation point of Individual and Team-up phases, which varies for different datasets.

Variant	Wiki-CS	Computers	Photo	Coauthor-CS
CSGCL–CAV–CED–T	77.68±0.34	88.29±0.11	92.52±0.34	92.50±0.08
CSGCL–CED–T	77.78±0.07	89.94±0.48	93.08±0.28	93.35±0.13
CSGCL–CAV–T	78.43±0.11	90.06±0.19	93.11±0.34	93.09±0.11
CSGCL–T	78.51±0.11	90.12±0.23	93.26±0.23	93.50±0.09
CSGCL-$\mathcal{S}$	77.88±0.15	89.52±0.26	92.85±0.34	93.41±0.07
CSGCL (Ours)	**78.60±0.13**	**90.17±0.17**	**93.32±0.21**	**93.55±0.12**

Table 5: Ablation study on node classification.

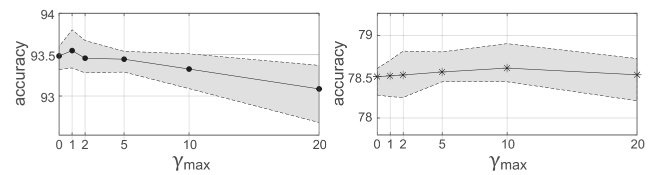
Furthermore, we visualize the graph representations of CSGCL as well as baselines by t-SNE [Van der Maaten and Hinton, 2008], a dimension reduction and visualization method. As shown in Figure 4, CSGCL learns the most discriminative graph representations among the competitive methods.

4.3 Ablation Studies

In this section, we describe the ablation studies on the key modules of CSGCL. "–CAV" and "–CED" refers to uniform attribute masking and edge dropping respectively, where the perturbation probabilities are set to the same for all attributes and edges. "–T" refers to the InfoNCE objective instead of Team-up. "–$\mathcal{S}$" refers to the disregard of community strength, where every community shares the average strength $\bar{\mathcal{S}}$.

Table 5 verifies the effectiveness of CSGCL. The classification accuracy over all 4 datasets increases with either CAV or CED, and gains over 1% improvement on average with both. CSGCL with Team-up objective reaches the best classification performance, bringing significant improvements over CSGCL–T on three datasets with the results $p = 1.37e$-2, $2.78e$-1, $3.71e$-2, and $1.37e$-2 of Wilcoxon signed-rank test in order. More importantly, the accuracy degrades without variations of community strength $\mathcal{S}$. This verifies that **the difference among community strength cannot be ignored**.

We also show that CSGCL has well preserved community semantics by carrying out community detection on the representations. Results are shown in Figure 5. The green curves (w/ CAV and CED) perform better than the red curves (w/ uniform attribute masking and edge dropping) after a period of training time, especially on Photo and Coauthor-CS; CSGCL, the blue curves using the Team-up objective, performs the best among its counterparts, with the performance gaps widening in the Team-up phase, especially on Computers and Photo. This shows that our community-strength-enhancement has better preserved community semantics throughout the learning process, which results in more accurate graph representations.



(a) Coauthor-CS (b) Wiki-CS

Figure 6: Node classification results with varied γ_{max}. The average accuracy scores are dotted, and the shaded area is the error band.

4.4 Parameter Analysis

In this section, we describe the parameter analysis on the upper bound of the strength coefficient, γ_{max}, which controls the overall influence of communities on a certain graph. All experiments below use the same training epochs and configurations. γ_{max} is set as 0, 1, 2, 5, 10, and 20. We take Coauthor-CS and Wiki-CS as examples.

As the results in Figure 6, the accuracy of node classification is relatively stable when γ_{max} changes in a certain range. So holistically, CSGCL is robust to the variation of γ_{max}, but it is recommended to use the most appropriate one for each dataset to achieve better performance. The classification accuracy will decrease if γ_{max} is too small (*e.g.* < 1 for Coauthor-CS), undermining the benefit of communities; however, it will also decrease if γ_{max} is too large (*e.g.* > 5 for Coauthor-CS), derailing the model training.

5 Conclusion

In this paper, we present CSGCL, a novel contrastive framework enhanced by community strength throughout the learning process. Firstly, we manage to preserve differences among communities by the enhanced augmentations on attributes and edges, CAV and CED. Secondly, we put forward the dynamic Team-up contrastive scheme which regards GCL as a social group activity, guiding the optimization with community strength in a progressive manner. CSGCL achieves state-of-the-art performance on three downstream tasks: node classification, node clustering, and link prediction, indicating the effectiveness and generality of community-strength-enhanced representations.

Acknowledgments

This work is supported by National Natural Science Foundation of China under grants U1936108, U1836204, 62206102, and Science and Technology Support Program of Hubei Province under grant 2022BAA046.

Contribution Statement

The paper is co-first authored by Han Chen and Ziwen Zhao.

References

[Ana and Jain, 2003] LN Fred Ana and Anil K Jain. Robust data clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 128–136, Madison, Wisconsin, June 2003. IEEE Computer Society.

[Becker and Hinton, 1992] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, January 1992.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, Virtual Event, Vienna, Austria, July 2020. PMLR.

[Girvan and Newman, 2002] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, June 2002.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, San Francisco, California, August 2016. Association for Computing Machinery.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4116–4126, Virtual Event, Vienna, Austria, July 2020. PMLR.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, Xi'an, China, July 2020. Association for Computing Machinery.

[Hjelm *et al.*, 2019] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May 2019.

[Khambatti *et al.*, 2002] Mujtaba Khambatti, Kyung Dong Ryu, and Partha Dasgupta. Peer-to-peer communities: formation and discovery. In *Proceedings of the Fourteenth IASTED International Conference on Parallel & Distributed Computing & Systems*, pages 161–166, Cambridge, Massachusetts, November 2002. ASTED/ACTA Press.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, April 2017.

[Lee *et al.*, 2022] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 7372–7380, Virtual Event, February 2022. AAAI Press.

[Li *et al.*, 2020] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *Proceeding of the 36th International Conference on Data Engineering (ICDE)*, pages 1677–1688, Dallas, Texas, April 2020. IEEE.

[Li *et al.*, 2022] Bolian Li, Baoyu Jing, and Hanghang Tong. Graph communal contrastive learning. In *Proceedings of the 31st ACM Web Conference 2022*, pages 1203–1213, Virtual Event, Lyon, France, April 2022. Association for Computing Machinery.

[Liu *et al.*, 2022] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip Yu. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[MacQueen, 1967] J MacQueen. Classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, California, 1967. University of California Press.

[Newman and Girvan, 2004] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, February 2004.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, New York, New York, August 2014. Association for Computing Machinery.

[Rao *et al.*, 2022] Jiahua Rao, Shuangjia Zheng, Sijie Mai, and Yuedong Yang. Communicative subgraph representation learning for multi-relational inductive drug-gene interaction prediction. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3919–3925, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization.

[Rong *et al.*, 2020] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *Proceedings of the 8th International Conference on Learning Representations*, Virtual Event, Addis Ababa, Ethiopia, April 2020.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *CoRR*, abs/1811.05868, 2018.

[Traag *et al.*, 2019] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):1–12, March 2019.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, November 2008.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, April 2018.

[Veličković *et al.*, 2019] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May 2019.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 203–209, San Francisco, California, February 2017. AAAI Press.

[Wei *et al.*, 2022] Chunyu Wei, Jian Liang, Di Liu, and Fei Wang. Contrastive graph structure learning via information bottleneck for recommendation. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, pages 20407–20420, New Orleans, Louisiana, November 2022. Curran Associates, Inc.

[Wilcoxon, 1992] Frank Wilcoxon. *Individual comparisons by ranking methods*. Springer, 1992.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pages 5812–5823, Virtual Event, Vancouver, Canada, December 2020. Curran Associates, Inc.

[Zachary, 1977] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, Winter 1977.

[Zhang *et al.*, 2022] Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang, and Jianzhong Qi. A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. *The VLDB Journal*, 31(5):1143–1168, May 2022.

[Zhao *et al.*, 2021] Yunxiang Zhao, Jianzhong Qi, Qingwei Liu, and Rui Zhang. WGCN: Graph convolutional networks with weighted structural features. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 624–633, Virtual Event, Montreal, Canada, July 2021.

[Zhou *et al.*, 2009] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, October 2009.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. In *ICML Workshop on Graph Representation Learning and Beyond*, Virtual Event, Vienna, Austria, July 2020. PMLR.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the 30th ACM Web Conference 2021*, pages 2069–2080, Ljubljana, Slovenia, April 2021. Association for Computing Machinery.

[Zhu *et al.*, 2022] Yun Zhu, Jianhao Guo, Fei Wu, and Siliang Tang. RoSA: A robust self-aligned framework for node-node graph contrastive learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3795–3801, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization.