Graph Contrastive Learning with Adaptive Augmentation

Yanqiao Zhu[1,2,*], Yichen Xu[3,*], Feng Yu[4], Qiang Liu[1,2], Shu Wu[1,2,†], and Liang Wang[1,2]
[1]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]School of Computer Science, Beijing University of Posts and Telecommunications [4]Alibaba Group
yanqiao.zhu@cripac.ia.ac.cn, linyxus@bupt.edu.cn
yf271406@alibaba-inc.com, {qiang.liu, shu.wu, wangliang}@nlpr.ia.ac.cn

ABSTRACT

Recently, contrastive learning (CL) has emerged as a successful method for unsupervised graph representation learning. Most graph CL methods first perform stochastic augmentation on the input graph to obtain two graph views and maximize the agreement of representations in the two views. Despite the prosperous development of graph CL methods, the design of graph augmentation schemes—a crucial component in CL—remains rarely explored. We argue that the data augmentation schemes should preserve intrinsic structures and attributes of graphs, which will force the model to learn representations that are insensitive to perturbation on unimportant nodes and edges. However, most existing methods adopt uniform data augmentation schemes, like uniformly dropping edges and uniformly shuffling features, leading to suboptimal performance. In this paper, we propose a novel graph contrastive representation learning method with adaptive augmentation that incorporates various priors for topological and semantic aspects of the graph. Specifically, on the topology level, we design augmentation schemes based on node centrality measures to highlight important connective structures. On the node attribute level, we corrupt node features by adding more noise to unimportant node features, to enforce the model to recognize underlying semantic information. We perform extensive experiments of node classification on a variety of real-world datasets. Experimental results demonstrate that our proposed method consistently outperforms existing state-of-the-art baselines and even surpasses some supervised counterparts, which validates the effectiveness of the proposed contrastive framework with adaptive augmentation.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; **Neural networks**; **Learning latent representations**.

KEYWORDS

Contrastive learning, graph representation learning, unsupervised learning, self-supervised learning

*The first two authors made equal contribution to this work.
†To whom correspondence should be addressed.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449802

ACM Reference Format:
Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449802

1 INTRODUCTION

Over the past few years, graph representation learning has emerged as a powerful strategy for analyzing graph-structured data. Graph representation learning using Graph Neural Networks (GNN) has received considerable attention, which aims to transform nodes to low-dimensional dense embeddings that preserve graph attributive and structural features. However, existing GNN models are mostly established in a supervised manner [20, 23, 45], which require abundant labeled nodes for training. Recently, Contrastive Learning (CL), as revitalization of the classical Information Maximization (InfoMax) principle [26], achieves great success in many fields, e.g., visual representation learning [1, 17, 41] and natural language processing [4, 28]. These CL methods seek to maximize the Mutual Information (MI) between the input (i.e. images) and its representations (i.e. image embeddings) by contrasting positive pairs with negative-sampled counterparts.

Inspired by previous CL methods, Deep Graph InfoMax (DGI) [46] marries the power of GNN into InfoMax-based methods. DGI firstly augments the original graph by simply shuffling node features. Then, a contrastive objective is proposed to maximize the MI between node embeddings and a global summary embedding. Following DGI, GMI [32] proposes two contrastive objectives to directly measure MI between input and representations of nodes and edges respectively, without explicit data augmentation. Moreover, to supplement the input graph with more global information, MVGRL [16] proposes to augment the input graph via graph diffusion kernels [24]. Then, it constructs graph views by uniformly sampling subgraphs and learns to contrast node representations to global embeddings across the two views.

Despite the prosperous development of graph CL methods, data augmentation schemes, proved to be a critical component for visual representation learning [48], remain rarely explored in existing literature. Unlike abundant data transformation techniques available for images and texts, graph augmentation schemes are non-trivial to define in CL methods, since graphs are far more complex due to the non-Euclidean property. We argue that the augmentation schemes used in the aforementioned methods suffer from two drawbacks. At first, simple data augmentation in *either* the structural domain *or* the attribute domain, such as feature shifting in DGI [46], is not sufficient for generating diverse neighborhoods (i.e.

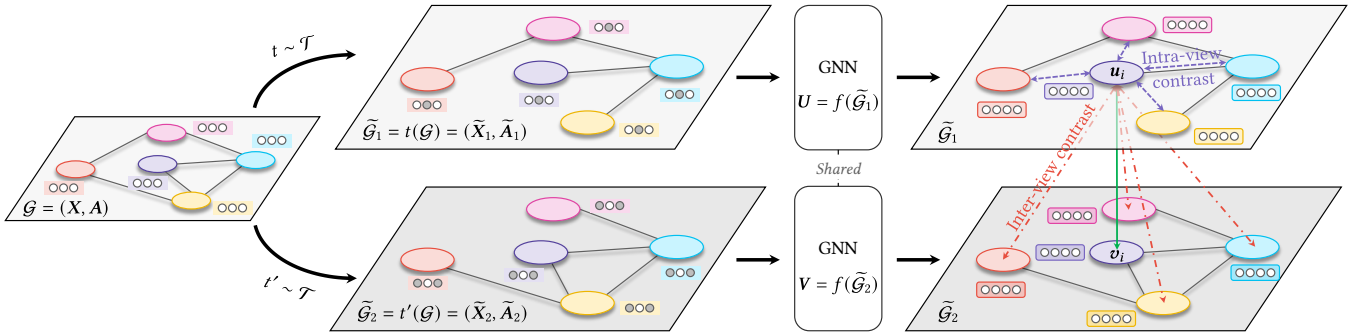


Figure 1: Our proposed deep Graph Contrastive representation learning with Adaptive augmentation (GCA) model. We first generate two graph views via stochastic augmentation that is adaptive to the graph structure and attributes. Then, the two graphs are fed into a shared Graph Neural Network (GNN) to learn representations. We train the model with a contrastive objective, which pulls representations of one node together while pushing node representations away from other node representations in the two views. N.B., we define the negative samples as all other nodes in the two views. Therefore, negative samples are from two sources, intra-view (in purple) and inter-view nodes (in red).

contexts) for nodes, especially when node features are sparse, leading to difficulty in optimizing the contrastive objective. Secondly, previous work ignores the discrepancy in the impact of nodes and edges when performing data augmentation. For example, if we construct graph views by *uniformly* dropping edges, removing some influential edges will deteriorate the embedding quality. As the representations learned by the contrastive objective tend to be *invariant* to corruption induced by the data augmentation scheme [50], the data augmentation strategies should be *adaptive* to the input graph to reflect its intrinsic patterns. Again, taking the edge removing scheme as an example, we can give larger probabilities to unimportant edges and lower probabilities to important ones, when randomly removing the edges. Then, this scheme is able to guide the model to ignore the introduced noise on unimportant edges and thus learn important patterns underneath the input graph.

To this end, we propose a novel contrastive framework for unsupervised graph representation learning, as shown in Figure 1, which we refer to as Graph Contrastive learning with Adaptive augmentation, GCA for brevity. In GCA, we first generate two correlated graph views by performing stochastic corruption on the input. Then, we train the model using a contrastive loss to maximize the agreement between node embeddings in these two views. Specifically, we propose a joint, adaptive data augmentation scheme at both topology and node attribute levels, namely removing edges and masking features, to provide diverse contexts for nodes in different views, so as to boost optimization of the contrastive objective. Moreover, we identify important edges and feature dimensions via centrality measures. Then, on the topology level, we adaptively drop edges by giving large removal probabilities to unimportant edges to highlight important connective structures. On the node attribute level, we corrupt attributes by adding more noise to unimportant feature dimensions, to enforce the model to recognize underlying semantic information.

The core contribution of this paper is two-fold:

- Firstly, we propose a general contrastive framework for unsupervised graph representation learning with strong, adaptive data augmentation. The proposed GCA framework jointly performs data augmentation on both topology and attribute levels that are adaptive to the graph structure and attributes, which encourages the model to learn important features from both aspects.
- Secondly, we conduct comprehensive empirical studies using five public benchmark datasets on node classification under the commonly-used linear evaluation protocol. GCA consistently outperforms existing methods and our unsupervised method even surpasses its supervised counterparts on several transductive tasks.

To make the results of this work reproducible, we make all the code publicly available at https://github.com/CRIPAC-DIG/GCA.

The remaining of the paper includes the following sections. We briefly review related work in Section 2. In Section 3, we present the proposed GCA model in detail. The results of the experiments are analyzed in Section 4. Finally, we conclude the paper in Section 5. For readers of interest, additional configurations of experiments and details of proofs are provided in Appendix A and B, respectively.

2 RELATED WORK

In this section, we briefly review prior work on contrastive representation learning. Then, we review graph representation learning methods. At last, we provide a summary of comparisons between the proposed method and its related work.

2.1 Contrastive Representation Learning

Being popular in self-supervised representation learning, contrastive methods aim to learn discriminative representations by contrasting positive and negative samples. For visual data, negative samples can be generated using a multiple-stage augmentation pipeline [1, 3, 6], consisting of color jitter, random flip, cropping, resizing, rotation [8], color distortion [25], etc. Existing work [17, 41, 49] employs a memory bank for storing negative samples. Other work [1, 3, 51] explores in-batch negative samples. For an image patch as the anchor, these methods usually find a global summary vector [1, 19] or patches in neighboring views [18, 44] as the positive sample, and

contrast them with negative-sampled counterparts, such as patches of other images within the same batch [19].

Theoretical analysis sheds light on the reasons behind their success [35]. Objectives used in these methods can be seen as maximizing a lower bound of MI between input features and their representations [26]. However, recent work [43] reveals that downstream performance in evaluating the quality of representations may strongly depend on the bias that is encoded not only in the convolutional architectures but also in the specific estimator of the InfoMax objective.

2.2 Graph Representation Learning

Many traditional methods on unsupervised graph representation learning inherently follow the contrastive paradigm [11, 14, 22, 34]. Prior work on unsupervised graph representation learning focuses on local contrastive patterns, which forces neighboring nodes to have similar embeddings. For example, in the pioneering work DeepWalk [34] and node2vec [11], nodes appearing in the same random walk are considered as positive samples. Moreover, to model probabilities of node co-occurrence pairs, many studies resort to Noise-Contrastive Estimation (NCE) [12]. However, these random-walk-based methods are proved to be equivalent to factorizing some forms of graph proximity (e.g., multiplication of the adjacent matrix to model high-order connection) [37] and thus tend to overly emphasize on the encoded structural information. Also, these methods are known to be error-prone with inappropriate hyperparameter tuning [11, 34].

Recent work on Graph Neural Networks (GNNs) employs more powerful graph convolutional encoders over conventional methods. Among them, considerable literature has grown up around the theme of supervised GNN [20, 23, 45, 47], which requires labeled datasets that may not be accessible in real-world applications. Along the other line of development, unsupervised GNNs receive little attention. Representative methods include GraphSAGE [15], which incorporates DeepWalk-like objectives. Recent work DGI [46] marries the power of GNN and CL, which focuses on maximizing MI between global graph-level and local node-level embeddings. Specifically, to implement the InfoMax objective, DGI requires an injective readout function to produce the global graph-level embedding. However, it is too restrictive to fulfill the injective property of the graph readout function, such that the graph embedding may be deteriorated. In contrast to DGI, our preliminary work [53] proposes to not rely on an explicit graph embedding, but rather focuses on maximizing the agreement of node embeddings across two corrupted views of the graph.

Following DGI, GMI [32] employs two discriminators to directly measure MI between input and representations of both nodes and edges without data augmentation; MVGRL [16] proposes to learn both node- and graph-level representations by performing node diffusion and contrasting node representations to augmented graph summary representations. Moreover, GCC [36] proposes a pretraining framework based on CL. It proposes to construct multiple graph views by sampling subgraphs based on random walks and then learn model weights with several feature engineering schemes. However, these methods do not explicitly consider adaptive graph augmentation at both structural and attribute levels, leading to suboptimal performance. Unlike these work, the adaptive data augmentation at both topology and attribute levels used in our GCA is able to preserve important patterns underneath the graph through stochastic perturbation.

Table 1: Comparison with related work.

Method	Contrastive objective	Topology	Attribute
DGI	Node–global	Uniform	—
GMI	Node–node	—	—
MVGRL	Node–global	Uniform	—
GCA	**Node–node**	**Adaptive**	**Adaptive**

Comparisons with related graph CL methods. In summary, we provide a brief comparison between the proposed GCA and other state-of-the-art graph contrastive representation learning methods, including DGI [46], GMI [32], and MVGRL [16] in Table 1, where the last two columns denote data augmentation strategies at topology and attribute levels respectively. It is seen that the proposed GCA method simplifies previous node–global contrastive scheme by defining contrastive objective at the node level. Most importantly, GCA is the only one that proposes adaptive data augmentation on both topology and attribute levels.

3 THE PROPOSED METHOD

In the following section, we present GCA in detail, starting with the overall contrastive learning framework, followed by the proposed adaptive graph augmentation schemes. Finally, we provide theoretical justification behind our method.

3.1 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the node set and the edge set respectively. We denote the feature matrix and the adjacency matrix as $\boldsymbol{X} \in \mathbb{R}^{N \times F}$ and $\boldsymbol{A} \in \{0, 1\}^{N \times N}$, where $\boldsymbol{x}_i \in \mathbb{R}^F$ is the feature of v_i, and $\boldsymbol{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. There is no given class information of nodes in $\mathcal{G}$ during training in the unsupervised setting. Our objective is to learn a GNN encoder $f(\boldsymbol{X}, \boldsymbol{A}) \in \mathbb{R}^{N \times F'}$ receiving the graph features and structure as input, that produces node embeddings in low dimensionality, i.e. $F' \ll F$. We denote $\boldsymbol{H} = f(\boldsymbol{X}, \boldsymbol{A})$ as the learned representations of nodes, where $\boldsymbol{h}_i$ is the embedding of node v_i. These representations can be used in downstream tasks, such as node classification and community detection.

3.2 The Contrastive Learning Framework

The proposed GCA framework follows the common graph CL paradigm where the model seeks to maximize the agreement of representations between different views [16, 53]. To be specific, we first generate two graph views by performing stochastic graph augmentation on the input. Then, we employ a contrastive objective that enforces the encoded embeddings of each node in the two different views to agree with each other and can be discriminated from embeddings of other nodes.

In our GCA model, at each iteration, we sample two stochastic augmentation function $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$, where $\mathcal{T}$ is the set of all possible augmentation functions. Then, we generate two graph views, denoted as $\widetilde{\mathcal{G}}_1 = t(\widetilde{\mathcal{G}})$ and $\widetilde{\mathcal{G}}_2 = t'(\widetilde{\mathcal{G}})$, and denote node embeddings in the two generated views as $\boldsymbol{U} = f(\widetilde{\boldsymbol{X}}_1, \widetilde{\boldsymbol{A}}_1)$ and $\boldsymbol{V} = f(\widetilde{\boldsymbol{X}}_2, \widetilde{\boldsymbol{A}}_2)$, where $\widetilde{\boldsymbol{X}}_*$ and $\widetilde{\boldsymbol{A}}_*$ are the feature matrices and adjacent matrices of the views.

After that, we employ a contrastive objective, i.e. a discriminator, that distinguishes the embeddings of the same node in these two different views from other node embeddings. For any node v_i, its embedding generated in one view, $\boldsymbol{u}_i$, is treated as the anchor, the embedding of it generated in the other view, $\boldsymbol{v}_i$, forms the positive sample, and the other embeddings in the two views are naturally regarded as negative samples. Mirroring the InfoNCE objective [44] in our multi-view graph CL setting, we define the pairwise objective for each positive pair $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ as

$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negative pairs}}}, \tag{1}$$

where τ is a temperature parameter. We define the critic $\theta(\boldsymbol{u}, \boldsymbol{v}) = s(g(\boldsymbol{u}), g(\boldsymbol{v}))$, where $s(\cdot, \cdot)$ is the cosine similarity and $g(\cdot)$ is a non-linear projection to enhance the expression power of the critic function [3, 43]. The projection function g in our method is implemented with a two-layer perceptron model.

Given a positive pair, we naturally define negative samples as all other nodes in the two views. Therefore, negative samples come from two sources, that are inter-view and intra-view nodes, corresponding to the second and the third term in the denominator in Eq. (1), respectively. Since two views are symmetric, the loss for another view is defined similarly for $\ell(\boldsymbol{v}_i, \boldsymbol{u}_i)$. The overall objective to be maximized is then defined as the average over all positive pairs, formally given by

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) + \ell(\boldsymbol{v}_i, \boldsymbol{u}_i) \right]. \tag{2}$$

To sum up, at each training epoch, GCA first draws two data augmentation functions t and t', and then generates two graph views $\widetilde{\mathcal{G}}_1 = t(\mathcal{G})$ and $\widetilde{\mathcal{G}}_2 = t'(\mathcal{G})$ of graph $\mathcal{G}$ accordingly. Then, we obtain node representations $\boldsymbol{U}$ and $\boldsymbol{V}$ of $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$ using a GNN encoder f. Finally, the parameters are updated by maximizing the objective in Eq. (2). The training algorithm is summarized in Algorithm 1.

3.3 Adaptive Graph Augmentation

In essence, CL methods that maximize agreement between views seek to learn representations that are *invariant* to perturbation introduced by the augmentation schemes [50]. In the GCA model, we propose to design augmentation schemes that tend to keep important structures and attributes unchanged, while perturbing possibly unimportant links and features. Specifically, we corrupt the input graph by randomly removing edges and masking node features in the graph, and the removing or masking probabilities are skewed for unimportant edges or features, that is, higher for unimportant edges or features, and lower for important ones. From an amortized perspective, we emphasize important structures and attributes over randomly corrupted views, which will guide the model to preserve fundamental topological and semantic graph patterns.

Algorithm 1: The GCA training algorithm

```
for epoch ← 1, 2, ··· do
    Sample two stochastic augmentation functions t ~ T and t' ~ T
    Generate two graph views G̃_1 = t(G) and G̃_2 = t'(G) by performing corruption on G
    Obtain node embeddings U of G̃_1 using the encoder f
    Obtain node embeddings V of G̃_2 using the encoder f
    Compute the contrastive objective J with Eq. (2)
    Update parameters by applying stochastic gradient ascent to maximize J
```

3.3.1 Topology-level augmentation. For topology-level augmentation, we consider a direct way for corrupting input graphs where we randomly remove edges in the graph [53]. Formally, we sample a modified subset $\widetilde{\mathcal{E}}$ from the original $\mathcal{E}$ with probability

$$P\{(u, v) \in \widetilde{\mathcal{E}}\} = 1 - p_{uv}^e, \tag{3}$$

where $(u, v) \in \mathcal{E}$ and p_{uv}^e is the probability of removing (u, v). $\widetilde{\mathcal{E}}$ is then used as the edge set in the generated view. p_{uv}^e should reflect the importance of the edge (u, v) such that the augmentation function are more likely to corrupt unimportant edges while keep important connective structures intact in augmented views.

In network science, node centrality is a widely-used measure that quantifies the influence of nodes in the graph [29]. We define edge centrality w_{uv}^e for edge (u, v) to measure its influence based on centrality of two connected nodes. Given a node centrality measure $\varphi_c(\cdot) : \mathcal{V} \to \mathbb{R}^+$, we define edge centrality as the average of two adjacent nodes' centrality scores, i.e. $w_{uv}^e = (\varphi_c(u) + \varphi_c(v))/2$, and on directed graph, we simply use the centrality of the tail node, i.e. $w_{uv}^e = \varphi_c(v)$, since the importance of edges is generally characterized by nodes they are pointing to [29].

Next, we calculate the probability of each edge based on its centrality value. Since node centrality values like degrees may vary across orders of magnitude [29], we first set $s_{uv}^e = \log w_{uv}^e$ to alleviate the impact of nodes with heavily dense connections. The probabilities can then be obtained after a normalization step that transform the values into probabilities, which is defined as

$$p_{uv}^e = \min\left(\frac{s_{\max}^e - s_{uv}^e}{s_{\max}^e - \mu_s^e} \cdot p_e, p_\tau \right), \tag{4}$$

where p_e is a hyperparameter that controls the overall probability of removing edges, $s_{\max}^e$ and μ_s^e is the maximum and average of s_{uv}^e, and $p_\tau < 1$ is a cut-off probability, used to truncate the probabilities since extremely high removal probabilities will lead to overly corrupted graph structures.

For the choice of the node centrality function, we use the following three centrality measures, including degree centrality, eigenvector centrality, and PageRank centrality due to their simplicity and effectiveness.

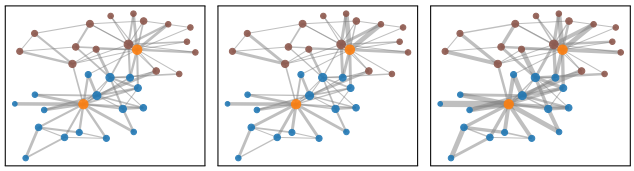
Degree centrality. Node degree itself can be a centrality measure [29]. On directed networks, we use in-degrees since the influence of a node in directed graphs are mostly bestowed by nodes pointing at it [29]. Despite that the node degree is one of the simplest centrality measures, it is quite effective and illuminating. For example, in citation networks where nodes represent papers and edges represent citation relationships, nodes with the highest degrees are likely to correspond to influential papers.

Eigenvector centrality. The eigenvector centrality [2, 29] of a node is calculated as its eigenvector corresponding to the largest eigenvalue of the adjacency matrix. Unlike degree centrality, which assumes that all neighbors contribute equally to the importance of the node, eigenvector centrality also takes the importance of neighboring nodes into consideration. By definition, the eigenvector centrality of each node is proportional to the sum of centralities of its neighbors, nodes that are either connected to many neighbors or connected to influential nodes will have high eigenvector centrality values. On directed graphs, we use the right eigenvector to compute the centrality, which corresponds to incoming edges. Note that since only the leading eigenvector is needed, the computational burden for calculating the eigenvector centrality is negligible.

PageRank centrality. The PageRank centrality [29, 30] is defined as the PageRank weights computed by the PageRank algorithm. The algorithm propagates influence along directed edges, and nodes gathered the most influence are regarded as important nodes. Formally, the centrality values are defined by

$$\boldsymbol{\sigma} = \alpha \boldsymbol{A} \boldsymbol{D}^{-1} \boldsymbol{\sigma} + \mathbf{1}, \tag{5}$$

where $\sigma \in \mathbb{R}^N$ is the vector of PageRank centrality scores for each node and α is a damping factor that prevents sinks in the graph from absorbing all ranks from other nodes connected to the sinks. We set $\alpha = 0.85$ as suggested in Page et al. [30]. For undirected graphs, we execute PageRank on transformed directed graphs, where each undirected edge is converted to two directed edges.



(a) Degree (b) Eigenvector (c) PageRank

Figure 2: Visualization of edge centrality computed by three schemes in the Karate club dataset, where centrality values are shown in terms of the thickness of edges. Node colors indicate two classes inside the network; two coaches are in orange.

To gain an intuition of these proposed adaptive structural augmentation schemes, we calculate edge centrality scores of the famous Karate club dataset [52], containing two groups of students leading by two coaches respectively. The edge centrality values calculated by different schemes are visualized in Figure 2. As can be seen in the figure, though the three schemes exhibit subtle differences, all of the augmentation schemes tend to emphasize edges that connect the two coaches (in orange) inside the two groups and put less attention to links between peripheral nodes across groups. This verifies that the proposed node-centrality-based adaptive topology augmentation scheme can recognize fundamental structures of the graph.

3.3.2 Node-attribute-level augmentation. On the node attribute level, similar to the salt-and-pepper noise in digital image processing [10], we add noise to node attributes via randomly masking a fraction of dimensions with zeros in node features. Formally, we first sample a random vector $\widetilde{\boldsymbol{m}} \in \{0,1\}^F$ where each dimension of it independently is drawn from a Bernoulli distribution independently, i.e., $\widetilde{m}_i \sim \text{Bern}(1 - p_i^f), \forall i$. Then, the generated node features $\widetilde{\boldsymbol{X}}$ is computed by

$$\widetilde{\boldsymbol{X}} = [\boldsymbol{x}_1 \circ \widetilde{\boldsymbol{m}}; \boldsymbol{x}_2 \circ \widetilde{\boldsymbol{m}}; \cdots; \boldsymbol{x}_N \circ \widetilde{\boldsymbol{m}}]^\top. \tag{6}$$

Here $[\cdot;\cdot]$ is the concatenation operator, and $\circ$ is the element-wise multiplication.

Similar to topology-level augmentation, the probability p_i^f should reflect the importance of the i-th dimension of node features. We assume that feature dimensions frequently appearing in influential nodes should be important, and define the weights of feature dimensions as follows. For sparse one-hot nodes features, i.e. $x_{ui} \in \{0,1\}$ for any node u and feature dimension i, we calculate the weight of dimension i as

$$w_i^f = \sum_{u \in \mathcal{V}} x_{ui} \cdot \varphi_c(u), \tag{7}$$

where $\varphi_c(\cdot)$ is a node centrality measure that is used to quantify node importance. The first term $x_{ui} \in \{0,1\}$ indicates the occurrence of dimension i in node u, and the second term $\varphi_i(u)$ measures the node importance of each occurrence. To provide some intuition behind the above definition, consider a citation network where each feature dimension corresponds to a keyword. Then, keywords that frequently appear in a highly influential paper should be considered informative and important.

For dense, continuous node features $\boldsymbol{x}_u$ of node u, where x_{ui} denotes feature value at dimension i, we cannot directly count the occurrence of each one-hot encoded value. Then, we turn to measure the magnitude of the feature value at dimension i of node u by its absolute value $|x_{ui}|$. Formally, we calculate the weights by

$$w_i^f = \sum_{u \in \mathcal{V}} |x_{ui}| \cdot \varphi_c(u). \tag{8}$$

Similar to topology augmentation, we perform normalization on the weights to obtain the probability representing feature importance. Formally,

$$p_i^f = \min\left(\frac{s_{\max}^f - s_i^f}{s_{\max}^f - \mu_s^f} \cdot p_f, p_\tau \right), \tag{9}$$

where $s_i^f = \log w_i^f$, $s_{\max}^f$ and μ_s^f is the maximum and the average value of s_i^f respectively, and p_f is a hyperparameter that controls the overall magnitude of feature augmentation.

Finally, we generate two corrupted graph views $\widetilde{\mathcal{G}}_1, \widetilde{\mathcal{G}}_2$ by jointly performing topology- and node-attribute-level augmentation. In GCA, the probability p_e and p_f is different for generating the two views to provide a diverse context for contrastive learning, where the probabilities for the first and the second view are denoted by $p_{e,1}, p_{f,1}$ and $p_{e,2}, p_{f,2}$ respectively.

In this paper, we propose and evaluate three model variants, denoted as GCA-DE, GCA-EV, and GCA-PR. The three variants employ degree, eigenvector, and PageRank centrality measures respectively. Note that all centrality and weight measures are only dependent on the topology and node attributes of the original graph. Therefore, they only need to be computed once and do not bring much computational burden.

3.4 Theoretical Justification

In this section, we provide theoretical justification behind our model from two perspectives, i.e. MI maximization and the triplet loss. Detailed proofs can be found in Appendix B.

Connections to MI maximization. Firstly, we reveal the connections between our loss and MI maximization between node features and the embeddings in the two views. The InfoMax principle has been widely applied in representation learning literature [1, 35, 41, 43]. MI quantifies the amount of information obtained about one random variable by observing the other random variable.

Theorem 1. *Let $\boldsymbol{X}_i = \{\boldsymbol{x}_k\}_{k\in\mathcal{N}(i)}$ be the neighborhood of node v_i that collectively maps to its output embedding, where $\mathcal{N}(i)$ denotes the set of neighbors of node v_i specified by GNN architectures, and $\boldsymbol{X}$ be the corresponding random variable with a uniform distribution $p(\boldsymbol{X}_i) = 1/N$. Given two random variables $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{F'}$ being the embedding in the two views, with their joint distribution denoted as $p(\boldsymbol{U}, \boldsymbol{V})$, our objective $\mathcal{J}$ is a lower bound of MI between encoder input $\boldsymbol{X}$ and node representations in two graph views $\boldsymbol{U}, \boldsymbol{V}$. Formally,*

$$\mathcal{J} \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V}). \tag{10}$$

Proof sketch. We first observe that our objective $\mathcal{J}$ is a lower bound of the InfoNCE objective [35, 44], defined by $I_{\text{NCE}}(\boldsymbol{U}; \boldsymbol{V}) \triangleq \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_j)}}\right]$. Since the InfoNCE estimator is a lower bound of the true MI, the theorem directly follows from the application of data processing inequality [5], which states that $I(\boldsymbol{U}; \boldsymbol{V}) \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V})$. □

Remark. Theorem 1 reveals that maximizing $\mathcal{J}$ is equivalent to explicitly maximizing a lower bound of the MI $I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V})$ between input node features and learned node representations. Recent work further provides empirical evidence that optimizing a stricter bound of MI may not lead to better downstream performance on visual representation learning [42, 43], which further highlights the importance of the design of data augmentation strategies.

When optimizing $I(\boldsymbol{U}; \boldsymbol{V})$, a lower bound of $I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V})$, we encourage the model to encode shared information between the two views. From the amortized perspective, corrupted views will follow a skewed distribution where important link structures and features are emphasized. By contrasting the two views, the model is enforced to encode the emphasized information into representations, which improves embedding quality.

However, as the objective is not defined specifically on negative samples generated by the augmentation function, it remains challenging to derive the relationship between specific augmentation functions and the lower bound. We shall leave it for future work.

Connections to the triplet loss. Alternatively, we may also view the optimization problem in Eq. (2) as a classical triplet loss, commonly used in deep metric learning.

Theorem 2. *When the projection function g is the identity function and we measure embedding similarity by simply taking the inner product, i.e. $s(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}$, and further assuming that positive pairs are far more aligned than negative pairs, i.e. $\boldsymbol{u}_i^\top \boldsymbol{v}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$, minimizing the pairwise objective $\ell(\boldsymbol{u}_i, \boldsymbol{v}_i)$ coincides with maximizing the triplet loss, as given in the sequel*

$$-\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) \propto 4N\tau + \sum_{j\neq i}\left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_j\|^2 + \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2\right). \tag{11}$$

Remark. Theorem 2 draws connections between the objective and the classical triplet loss. In other words, we may regard the problem in Eq. (2) as learning graph convolutional encoders to encourage positive samples being further away from negative samples in the embedding space. Moreover, by viewing the objective from the metric learning perspective, we highlight the importance of appropriate data augmentation schemes, which is often neglected in previous InfoMax-based methods. Specifically, as the objective pulls together representation of each node in the two corrupted views, the model is enforced to encode information in the input graph that is insensitive to perturbation. Since the proposed adaptive augmentation schemes tend to keep important link structures and node attributes intact in the perturbation, the model is guided to encode essential structural and semantic information into the representation, which improves the quality of embeddings. Last, the contrastive objective used in GCA is cheap to optimize, since we do not have to generate negative samples explicitly and all computation can be performed in parallel. In contrast, the triplet loss is known to be computationally expensive [38].

4 EXPERIMENTS

In this section, we conduct experiments to evaluate our model through answering the following questions.

- **RQ1**. Does our proposed GCA outperform existing baseline methods on node classification?
- **RQ2**. Do all proposed adaptive graph augmentation schemes benefit the learning of the proposed model? How does each graph augmentation scheme affect model performance?
- **RQ3**. Is the proposed model sensitive to hyperparameters? How do key hyperparameters impact the model performance?

We begin with a brief introduction of the experimental setup, and then we proceed to details of experimental results and their analysis.

Table 2: Statistics of datasets used in experiments.

Dataset	#Nodes	#Edges	#Features	#Classes
Wiki-CS[1]	11,701	216,123	300	10
Amazon-Computers[2]	13,752	245,861	767	10
Amazon-Photo[3]	7,650	119,081	745	8
Coauthor-CS[4]	18,333	81,894	6,805	15
Coauthor-Physics[5]	34,493	247,962	8,415	5

[1] https://github.com/pmernyei/wiki-cs-dataset/raw/master/dataset
[2] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_computers.npz
[3] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_photo.npz
[4] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/ms_academic_cs.npz
[5] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/ms_academic_phy.npz

4.1 Experimental Setup

4.1.1 Datasets. For comprehensive comparison, we use six widely-used datasets, including Wiki-CS, Amazon-Computers, Amazon-Photo, Coauthor-CS, and Coauthor-Physics, to study the performance of transductive node classification. The datasets are collected from real-world networks from different domains; their detailed statistics is summarized in Table 2.

- **Wiki-CS** [27] is a reference network constructed based on Wikipedia. The nodes correspond to articles about computer science and edges are hyperlinks between the articles. Nodes are labeled with ten classes each representing a branch of the field. Node features are calculated as the average of pre-trained GloVe [33] word embeddings of words in each article.
- **Amazon-Computers** and **Amazon-Photo** [39] are two networks of co-purchase relationships constructed from Amazon, where nodes are goods and two goods are connected when they are frequently bought together. Each node has a sparse bag-of-words feature encoding product reviews and is labeled with its category.
- **Coauthor-CS** and **Coauthor-Physics** [39] are two academic networks, which contain co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge. In these graphs, nodes represent authors and edges indicate co-authorship relationships; that is, two nodes are connected if they have co-authored a paper. Each node has a sparse bag-of-words feature based on paper keywords of the author. The label of an author corresponds to their most active research field.

Among these datasets, Wiki-CS has dense numerical features, while the other four datasets only contain sparse one-hot features. For the Wiki-CS dataset, we evaluate the models on the public splits shipped with the dataset [27]. Regarding the other four datasets, since they have no public splits available, we instead randomly split the datasets, where 10%, 10%, and the rest 80% of nodes are selected for the training, validation, and test set, respectively.

4.1.2 Evaluation protocol. For every experiment, we follow the linear evaluation scheme as introduced in Veličković et al. [46], where each model is firstly trained in an unsupervised manner; then, the resulting embeddings are used to train and test a simple ℓ_2-regularized logistic regression classifier. We train the model for twenty runs for different data splits and report the averaged performance on each dataset for fair evaluation. Moreover, we measure performance in terms of accuracy in these experiments.

4.1.3 Baselines. We consider representative baseline methods belonging to the following two categories: (1) traditional methods including DeepWalk [34] and node2vec [11] and (2) deep learning methods including Graph Autoencoders (GAE, VGAE) [22], Deep Graph Infomax (DGI) [46], Graphical Mutual Information Maximization (GMI) [32], and Multi-View Graph Representation Learning (MVGRL) [16]. Furthermore, we report the performance obtained using a logistic regression classifier on raw node features and DeepWalk with embeddings concatenated with input node features. To directly compare our proposed method with supervised counterparts, we also report the performance of two representative models Graph Convolutional Networks (GCN) [23] and Graph Attention Networks (GAT) [45], where they are trained in an end-to-end fashion. For all baselines, we report their performance based on their official implementations.

4.1.4 Implementation details. We employ a two-layer GCN [23] as the encoder for all deep learning baselines due to its simplicity. The encoder architecture is formally given by

$$\mathrm{GC}_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma\left(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}_i\right), \tag{12}$$

$$f(\boldsymbol{X}, \boldsymbol{A}) = \mathrm{GC}_2(\mathrm{GC}_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}). \tag{13}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with self-loops, $\hat{\boldsymbol{D}} = \sum_i \hat{\boldsymbol{A}}_i$ is the degree matrix, $\sigma(\cdot)$ is a nonlinear activation function, e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$, and $\boldsymbol{W}_i$ is a trainable weight matrix. For experimental specifications, including details of the configurations of the optimizer and hyperparameter settings, we refer readers of interest to Appendix A.

4.2 Performance on Node Classification (RQ1)

The empirical performance is summarized in Table 3. Overall, from the table, we can see that our proposed model shows strong performance across all five datasets. GCA consistently performs better than unsupervised baselines by considerable margins on both transductive tasks. The strong performance verifies the superiority of the proposed contrastive learning framework. On the two Coauthor datasets, we note that existing baselines have already obtained high enough performance; our method GCA still pushes that boundary forward. Moreover, we particularly note that GCA is competitive with models *trained with label supervision* on all five datasets.

We make other observations as follows. Firstly, the performance of traditional contrastive learning methods like DeepWalk is inferior to the simple logistic regression classifier that only uses raw features on some datasets (Coauthor-CS and Coauthor-Physics), which suggests that these methods may be ineffective in utilizing node features. Unlike traditional work, we see that GCN-based methods, e.g., GAE, are capable of incorporating node features when learning embeddings. However, we note that on certain datasets (Wiki-CS), their performance is still worse than DeepWalk + feature, which we believe can be attributed to their naïve method of selecting negative samples that simply chooses contrastive pairs based on edges. This fact further demonstrates the important role of selecting negative

Table 3: Summary of performance on node classification in terms of accuracy in percentage with standard deviation. Available data for each method during the training phase is shown in the second column, where X, A, Y correspond to node features, the adjacency matrix, and labels respectively. The highest performance of unsupervised models is highlighted in boldface; the highest performance of supervised models is underlined. OOM indicates Out-Of-Memory on a 32GB GPU.

Method	Training Data	Wiki-CS	Amazon-Computers	Amazon-Photo	Coauthor-CS	Coauthor-Physics
Raw features	$\boldsymbol{X}$	71.98 ± 0.00	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00
node2vec	$\boldsymbol{A}$	71.79 ± 0.05	84.39 ± 0.08	89.67 ± 0.12	85.08 ± 0.03	91.19 ± 0.04
DeepWalk	$\boldsymbol{A}$	74.35 ± 0.06	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15
DeepWalk + features	$\boldsymbol{X}, \boldsymbol{A}$	77.21 ± 0.03	86.28 ± 0.07	90.05 ± 0.08	87.70 ± 0.04	94.90 ± 0.09
GAE	$\boldsymbol{X}, \boldsymbol{A}$	70.15 ± 0.01	85.27 ± 0.19	91.62 ± 0.13	90.01 ± 0.71	94.92 ± 0.07
VGAE	$\boldsymbol{X}, \boldsymbol{A}$	75.63 ± 0.19	86.37 ± 0.21	92.20 ± 0.11	92.11 ± 0.09	94.52 ± 0.00
DGI	$\boldsymbol{X}, \boldsymbol{A}$	75.35 ± 0.14	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52
GMI	$\boldsymbol{X}, \boldsymbol{A}$	74.85 ± 0.08	82.21 ± 0.31	90.68 ± 0.17	OOM	OOM
MVGRL	$\boldsymbol{X}, \boldsymbol{A}$	77.52 ± 0.08	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03
GCA-DE	$\boldsymbol{X}, \boldsymbol{A}$	78.30 ± 0.00	**87.85 ± 0.31**	92.49 ± 0.09	**93.10 ± 0.01**	95.68 ± 0.05
GCA-PR	$\boldsymbol{X}, \boldsymbol{A}$	**78.35 ± 0.05**	87.80 ± 0.23	**92.53 ± 0.16**	93.06 ± 0.03	95.72 ± 0.03
GCA-EV	$\boldsymbol{X}, \boldsymbol{A}$	78.23 ± 0.04	87.54 ± 0.49	92.24 ± 0.21	92.95 ± 0.13	**95.73 ± 0.03**
GCN	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	77.19 ± 0.12	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	95.65 ± 0.16
GAT	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	77.65 ± 0.11	86.93 ± 0.29	92.56 ± 0.35	92.31 ± 0.24	95.47 ± 0.15

Table 4: Performance of model variants on node classification in terms of accuracy in percentage with standard deviation. We use the degree centrality in all variants. The highest performance is highlighted in boldface.

Variant	Topology	Attribute	Wiki-CS	Amazon-Computers	Amazon-Photo	Coauthor-CS	Coauthor-Physics
GCA–T–A	Uniform	Uniform	78.19 ± 0.01	86.25 ± 0.25	92.15 ± 0.24	92.93 ± 0.01	95.26 ± 0.02
GCA–T	Uniform	Adaptive	78.23 ± 0.02	86.72 ± 0.49	92.20 ± 0.26	93.07 ± 0.01	95.59 ± 0.04
GCA–A	Adaptive	Uniform	78.25 ± 0.02	87.66 ± 0.30	92.23 ± 0.20	93.02 ± 0.01	95.54 ± 0.02
GCA	Adaptive	Adaptive	**78.30 ± 0.01**	**87.85 ± 0.31**	**92.49 ± 0.09**	**93.10 ± 0.01**	**95.68 ± 0.05**

samples based on augmented graph views in contrastive representation learning. Moreover, compared to existing baselines DGI, GMI, and MVGRL, our proposed method performs strong, adaptive data augmentation in constructing negative samples, leading to better performance. Note that, although MVGRL employs diffusion to incorporate global information into augmented views, it still fails to consider the impacts of different edges adaptively on input graphs. The superior performance of GCA verifies that our proposed adaptive data augmentation scheme is able to help improve embedding quality by preserving important patterns during perturbation.

Secondly, we observe that all three variants with different node centrality measures of GCA outperform existing contrastive baselines on all datasets. We also notice that GCA-DE and GCA-PR with the degree and PageRank centrality respectively are two strong variants that achieve the best or competitive performance on all datasets. Please kindly note that the result indicates that our model is not limited to specific choices of centrality measures and verifies the effectiveness and generality of our proposed framework.

In summary, the superior performance of GCA compared to existing state-of-the-art methods verifies the effectiveness of our proposed GCA framework that performs data augmentation adaptive to the graph structure and attributes.

4.3 Ablation Studies (RQ2)

In this section, we substitute the proposed topology and attribute level augmentation with their uniform counterparts to study the impact of each component of GCA. GCA–T–A denotes the model with uniform topology and node attribute augmentation schemes, where the probabilities of dropping edge and masking features are set to the same for all nodes. The variants GCA–T and GCA–A are defined similarly except that we substitute the topology and the node attribute augmentation scheme with uniform sampling in the two models respectively. Degree centrality is used in all the variants for fair comparison. Please kindly note that the downgraded GCA–T–A fallbacks to our preliminary work GRACE [53].

The results are presented in Table 4, where we can see that both topology-level and node-attribute-level adaptive augmentation scheme improve model performance consistently on all datasets. In addition, the combination of adaptive augmentation schemes on the two levels further benefits the performance. On the Amazon-Computers dataset, our proposed GCA gains 1.5% absolute improvement compared to the base model with no adaptive augmentation enabled. The results verify the effectiveness of our adaptive augmentation schemes on both topology and node attribute levels.

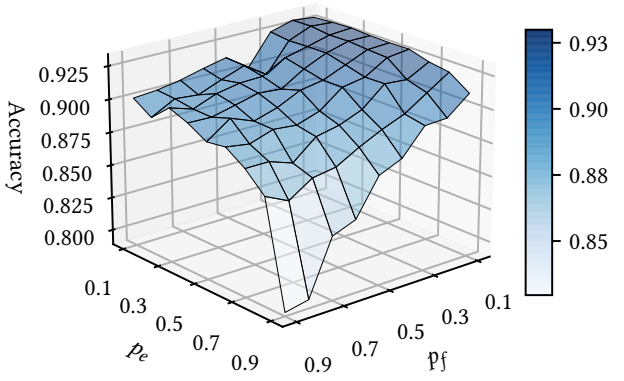


Figure 3: The performance of GCA with varied different hyperparameters on the Amazon-Photo dataset in terms of node classification accuracy.

4.4 Sensitivity Analysis (RQ3)

In this section, we perform sensitivity analysis on critical hyperparameters in GCA, namely four probabilities $p_{e,1}, p_{f,1}, p_{e,2}$, and $p_{f,2}$ that determine the generation of graph views to show the stability of the model under perturbation of these hyperparameters. We conduct transductive node classification by varying these parameters from 0.1 to 0.9. For sake of visualization brevity, we set $p_e = p_{e,1} = p_{e,2}$ and $p_f = p_{f,1} = p_{f,2}$ to control the magnitude of the proposed topology and node attribute level augmentation. We only change these four parameters in the sensitivity analysis, and other parameters remain the same as previously described.

The results on the Amazon-Photo dataset are shown in Figure 3. From the figure, it can be observed that the performance of node classification in terms of accuracy is relatively stable when the parameters are not too large, as shown in the plateau in the figure. We thus conclude that, overall, our model is insensitive to these probabilities, demonstrating the robustness to hyperparameter perturbation. If the probability is set too large (e.g., > 0.5), the original graph will be heavily undermined. For example, when $p_e = 0.9$, almost every existing edge has been removed, leading to isolated nodes in the generated graph views. Under such circumstances, the GNN is hard to learn useful information from node neighborhoods. Therefore, the learned node embeddings in the two graph views are not distinctive enough, which will result in the difficulty of optimizing the contrastive objective.

5 CONCLUSION

In this paper, we have developed a novel graph contrastive representation learning framework with adaptive augmentation. Our model learns representation by maximizing the agreement of node embeddings between views that are generated by adaptive graph augmentation. The proposed adaptive augmentation scheme first identifies important edges and feature dimensions via network centrality measures. Then, on the topology level, we randomly remove edges by assigning large probabilities on unimportant edges to enforce the model to recognize network connectivity patterns. On the node attribute level, we corrupt attributes by adding more noise to unimportant feature dimensions to emphasize the underlying semantic information. We have conducted comprehensive experiments using various real-world datasets. Experimental results demonstrate that our proposed GCA method consistently outperforms existing state-of-the-art methods and even surpasses several supervised counterparts.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their insightful comments. The authors would also like to thank Mr. Tao Sun and Mr. Sirui Lu for their valuable discussion. This work is jointly supported by National Natural Science Foundation of China (U19B2038, 61772528) and Beijing National Natural Science Foundation (4182066).

DISCUSSIONS ON BROADER IMPACT

This paper presents a novel graph contrastive learning framework, and we believe it would be beneficial to the graph machine learning community both theoretically and practically. Our proposed self-supervised graph representation learning techniques help alleviate the label scarcity issue when deploying machine learning applications in real-world, which saves a lot of efforts on human annotating. For example, our GCA framework can be plugged into existing recommender systems and produces high-quality embeddings for users and items to resolve the cold start problem. Note that our work mainly serves as a plug-in for existing machine learning models, it does not bring new ethical concerns. However, the GCA model may still give biased outputs (e.g., gender bias, ethnicity bias), as the provided data itself may be strongly biased during the processes of the data collection, graph construction, etc.

A IMPLEMENTATION DETAILS

A.1 Computing Infrastructures

Software infrastructures. All models are implemented using PyTorch Geometric 1.6.1 [7], PyTorch 1.6.0 [31], and NetworkX 2.5 [13]. All datasets used throughout experiments are available in PyTorch Geometric libraries.

Hardware infrastructures. We conduct experiments on a computer server with four NVIDIA Tesla V100S GPUs (with 32GB memory each) and twelve Intel Xeon Silver 4214 CPUs.

A.2 Hyperparameter Specifications

All model parameters are initialized with Glorot initialization [9], and trained using Adam SGD optimizer [21] on all datasets. The ℓ_2 weight decay factor is set to 10^{-5} and the dropout rate [40] is set to zero on all datasets. The probability parameters controlling the sampling process, $p_{e,1}, p_{f,1}$ for the first view and $p_{e,2}, p_{f,2}$ for the second view, are all selected between 0.0 and 0.4, since the original graph will be overly corrupted when the probability is set too large. Note that to generate different contexts for nodes in the two views, $p_{e,1}$ and $p_{e,2}$ should be distinct, and the same holds for $p_{f,1}$ and $p_{f,2}$. All dataset-specific hyperparameter configurations are summarized in Table 5.

Table 5: Hypeparameter specifications.

Dataset	$p_{e,1}$	$p_{e,2}$	$p_{f,1}$	$p_{f,2}$	p_τ	τ	Learning rate	Training epochs	Hidden dimension	Activation function
Wiki-CS	0.2	0.4	0.1	0.1	0.7	0.6	0.01	3,000	256	PReLU
Amazon-Computers	0.5	0.5	0.2	0.1	0.7	0.1	0.01	1,500	128	PReLU
Amazon-Photo	0.3	0.5	0.1	0.1	0.7	0.3	0.1	2,000	256	ReLU
Coauthor-CS	0.3	0.2	0.3	0.4	0.7	0.4	0.0005	1,000	256	RReLU
Coauthor-Physics	0.4	0.1	0.1	0.4	0.7	0.5	0.01	1,500	128	RReLU

B DETAILED PROOFS

B.1 Proof of Theorem 1

Theorem 1. *Let $\boldsymbol{X}_i = \{\boldsymbol{x}_k\}_{k\in\mathcal{N}(i)}$ be the neighborhood of node v_i that collectively maps to its output embedding, where $\mathcal{N}(i)$ denotes the set of neighbors of node v_i specified by GNN architectures, and $\boldsymbol{X}$ be the corresponding random variable with a uniform distribution $p(\boldsymbol{X}_i) = 1/N$. Given two random variables $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{F'}$ being the embedding in the two views, with their joint distribution denoted as $p(\boldsymbol{U}, \boldsymbol{V})$, our objective $\mathcal{J}$ is a lower bound of MI between encoder input $\boldsymbol{X}$ and node representations in two graph views $\boldsymbol{U}, \boldsymbol{V}$. Formally,*

$$\mathcal{J} \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V}). \tag{14}$$

Proof. We first show the connection between our objective $\mathcal{J}$ and the InfoNCE objective [35, 44] , which is defined as

$$I_{\mathrm{NCE}}(\boldsymbol{U}; \boldsymbol{V}) \triangleq \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)} \left[\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)}}{\frac{1}{N} \sum_{j=1}^{N} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_j)}} \right],$$

where the critic function is defined as $\theta(\boldsymbol{x}, \boldsymbol{y}) = s(g(\boldsymbol{x}), g(\boldsymbol{y}))$. We further define $\rho_r(\boldsymbol{u}_i) = \sum_{j\neq i}^{N} \exp(\theta(\boldsymbol{u}_i, \boldsymbol{u}_j)/\tau)$ and $\rho_c(\boldsymbol{u}_i) = \sum_{j=1}^{N} \exp(\theta(\boldsymbol{u}_i, \boldsymbol{v}_j)/\tau)$ for convenience of notation. $\rho_r(\boldsymbol{v}_i)$ and $\rho_c(\boldsymbol{v}_i)$ can be defined symmetrically. Then, our objective $\mathcal{J}$ can be rewritten as

$$\mathcal{J} = \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)} \left[\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau)}{\sqrt{(\rho_c(\boldsymbol{u}_i) + \rho_r(\boldsymbol{u}_i))\,(\rho_c(\boldsymbol{v}_i) + \rho_r(\boldsymbol{v}_i))}} \right]. \tag{15}$$

Using the notation of ρ_c, the InfoNCE estimator I_{NCE} can be written as

$$I_{\mathrm{NCE}}(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)} \left[\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau)}{\rho_c(\boldsymbol{u}_i)} \right]. \tag{16}$$

Therefore,

$$\begin{aligned} 2\mathcal{J} &= I_{\mathrm{NCE}}(\boldsymbol{U}, \boldsymbol{V}) - \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)} \left[\frac{1}{N} \sum_{i=1}^{N} \log \left(1 + \frac{\rho_r(\boldsymbol{u}_i)}{\rho_c(\boldsymbol{u}_i)} \right) \right] \\ &\quad + I_{\mathrm{NCE}}(\boldsymbol{V}, \boldsymbol{U}) - \mathbb{E}_{\prod_i p(\boldsymbol{u}_i, \boldsymbol{v}_i)} \left[\frac{1}{N} \sum_{i=1}^{N} \log \left(1 + \frac{\rho_r(\boldsymbol{v}_i)}{\rho_c(\boldsymbol{v}_i)} \right) \right] \\ &\le I_{\mathrm{NCE}}(\boldsymbol{U}, \boldsymbol{V}) + I_{\mathrm{NCE}}(\boldsymbol{V}, \boldsymbol{U}). \end{aligned} \tag{17}$$

According to Poole et al. [35], the InfoNCE estimator is a lower bound of the true MI, i.e.

$$I_{\mathrm{NCE}}(\boldsymbol{U}, \boldsymbol{V}) \le I(\boldsymbol{U}; \boldsymbol{V}). \tag{18}$$

Thus, we arrive at

$$2\mathcal{J} \le I(\boldsymbol{U}; \boldsymbol{V}) + I(\boldsymbol{V}; \boldsymbol{U}) = 2I(\boldsymbol{U}; \boldsymbol{V}), \tag{19}$$

which leads to the inequality

$$\mathcal{J} \le I(\boldsymbol{U}; \boldsymbol{V}). \tag{20}$$

According to the data processing inequality [5], which states that, for all random variables $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ satisfying the Markov relation $\boldsymbol{X} \to \boldsymbol{Y} \to \boldsymbol{Z}$, the inequality $I(\boldsymbol{X}; \boldsymbol{Z}) \le I(\boldsymbol{X}; \boldsymbol{Y})$ holds. Then, we observe that $\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{V}$ satisfy the relation $\boldsymbol{U} \leftarrow \boldsymbol{X} \to \boldsymbol{V}$. Since, $\boldsymbol{U}$ and $\boldsymbol{V}$ are conditionally independent after observing $\boldsymbol{X}$, the relation is Markov equivalent to $\boldsymbol{U} \to \boldsymbol{X} \to \boldsymbol{V}$, which leads to $I(\boldsymbol{U}; \boldsymbol{V}) \le I(\boldsymbol{U}; \boldsymbol{X})$. We further notice that the relation $\boldsymbol{X} \to (\boldsymbol{U}, \boldsymbol{V}) \to \boldsymbol{U}$ holds, and hence it follows that $I(\boldsymbol{X}; \boldsymbol{U}) \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V})$. Combining the two inequalities yields the required inequality

$$I(\boldsymbol{U}; \boldsymbol{V}) \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V}). \tag{21}$$

Following Eq. (20) and Eq. (21), we finally arrive at inequality

$$\mathcal{J} \le I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V}), \tag{22}$$

which concludes the proof. □

B.2 Proof of Theorem 2

Theorem 2. *When the projection function g is the identity function and we measure embedding similarity by simply taking inner product, and further assuming that positive pairs are far more aligned than negative pairs, i.e. $\boldsymbol{u}_i^\top \boldsymbol{v}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$, minimizing the pairwise objective $\ell(\boldsymbol{u}_i, \boldsymbol{v}_i)$ coincides with maximizing the triplet loss, as given in the sequel*

$$\begin{aligned} -\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) \propto 4N\tau + \sum_{j\neq i} \Big[&\left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_j\|^2 \right) \\ &+ \left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 \right) \Big]. \end{aligned} \tag{23}$$

Proof. Based on the assumptions, we can rearrange the pairwise objective as

$$\begin{aligned} -\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) &= -\log \frac{e^{(\boldsymbol{u}_i^\top \boldsymbol{v}_i/\tau)}}{\sum_{k=1}^{N} e^{(\boldsymbol{u}_i^\top \boldsymbol{v}_k/\tau)} + \sum_{k\neq i}^{N} e^{(\boldsymbol{u}_i^\top \boldsymbol{u}_k/\tau)}} \\ &= \log \left(1 + \sum_{k\neq i}^{N} e^{\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau} \right)} + \sum_{k\neq i}^{N} e^{\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau} \right)} \right). \end{aligned} \tag{24}$$

By Taylor expansion of first order,

$$
\begin{aligned}
&-\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) \\
&\approx \sum_{k\neq i}^{N} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right) + \sum_{k\neq i}^{N} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right) \\
&\approx 2 + \frac{1}{\tau}\left[\sum_{k\neq i}^{N} (\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i) + \sum_{k\neq i}^{N} (\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i)\right] \\
&= 2 - \frac{1}{2\tau}\sum_{k\neq i}^{N}\left(\|\boldsymbol{u}_i - \boldsymbol{v}_k\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 + \|\boldsymbol{u}_i - \boldsymbol{u}_k\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2\right) \\
&\propto 4N\tau + \sum_{k\neq i}^{N}\left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_k\|^2 + \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_k\|^2\right),
\end{aligned}
\tag{25}
$$

which concludes the proof. □

REFERENCES

[1] Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning Representations by Maximizing Mutual Information Across Views. In *Advances in Neural Information Processing Systems 32*. 15509–15519.

[2] Phillip Bonacich. 1987. Power and Centrality: A Family of Measures. *Amer. J. Sociology* 92, 5 (March 1987), 1170–1182.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119. PMLR, 10709–10719.

[4] Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM Press, 160–167.

[5] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Second Edition)*. Wiley-Interscience, USA.

[6] William Falcon and Kyunghyun Cho. 2020. A Framework For Contrastive Self-Supervised Learning and Designing A New Approach. *arXiv.org* (Sept. 2020). arXiv:2009.00104v1 [cs.CV]

[7] Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *Proceedings of the 6th International Conference on Learning Representations*.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR.org, 249–256.

[10] Rafael C. Gonzalez and Richard E. Woods. 2018. *Digital Image Processing (Fourth Edition)*. Pearson, USA.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[12] Michael Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research* 13 (2012), 307–361.

[13] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*. 11–15.

[14] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 40, 3 (2017), 52–74.

[15] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*. 1024–1034.

[16] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 3451–3461.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 9726–9735.

[18] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. 2020. Data-Efficient Image Recognition with Contrastive Predictive Coding. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4182–4192.

[19] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning Deep Representations by Mutual Information Estimation and Maximization. In *Proceedings of the 7th International Conference on Learning Representations*.

[20] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. 2019. Hierarchical Graph Convolutional Networks for Semi-supervised Node Classification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. IJCAI.org, 4532–4539.

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

[22] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. In *Bayesian Deep Learning Workshop@NIPS*.

[23] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.

[24] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems 32*. 13333–13345.

[25] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2017. Colorization as a Proxy Task for Visual Understanding. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 840–849.

[26] Ralph Linsker. 1988. Self-Organization in a Perceptual Network. *IEEE Computer* 21, 3 (1988), 105–117.

[27] Péter Mernyei and Catalina Cangea. 2020. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. In *ICML Workshop on Graph Representation Learning and Beyond*.

[28] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning Word Embeddings Efficiently with Noise-Contrastive Estimation. In *Advances in Neural Information Processing Systems 26*. 2265–2273.

[29] Mark E. J. Newman. 2018. *Networks: An Introduction (Second Edition)*. Oxford University Press.

[30] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 8024–8035.

[32] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *Proceedings of the Web Conference 2020*. ACM, 259–270.

[33] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. ACL, 1532–1543.

[34] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710.

[35] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. 2019. On Variational Bounds of Mutual Information. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 5171–5180.

[36] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1150–1160.

[37] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 459–467.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 815–823.

[39] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *arXiv.org* (Nov. 2018). arXiv:1811.05868v2 [cs.LG]

[40] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[41] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive Multiview Coding. *arXiv.org* (June 2019). arXiv:1906.05849v4 [cs.CV]

[42] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What Makes for Good Views for Contrastive Learning. *arXiv.org* (May 2020). arXiv:2005.10243v1 [cs.CV]

[43] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. 2020. On Mutual Information Maximization for Representation Learning. In *Proceedings of the 8th International Conference on Learning Representations*.

[44] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv.org* (2018). arXiv:1807.03748v2 [cs.LG]

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations*.

[46] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *Proceedings of the 7th International Conference on Learning Representations*.

[47] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6861–6871.

[48] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. 2020. On Mutual Information in Contrastive Learning for Visual Representations. *arXiv.org* (May 2020). arXiv:2005.13149v2 [cs.LG]

[49] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3733–3742.

[50] Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. 2020. What Should Not Be Contrastive in Contrastive Learning. *arXiv.org* (Aug. 2020). arXiv:2008.05659v1 [cs.CV]

[51] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. 2019. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6210–6219.

[52] Wayne W. Zachary. 1977. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research* 33, 4 (1977), 452–473.

[53] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. In *ICML Workshop on Graph Representation Learning and Beyond*.