Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning

Kaize Ding*, Yancheng Wang*, Yingzhen Yang, and Huan Liu
Arizona State University
School of Computing and Augmented Intelligence
kaize.ding@asu.edu, yancheng.wang@asu.edu, yingzhen.yang@asu.edu, huan.liu@asu.edu

Abstract

Graph Contrastive Learning (GCL) has recently drawn much research interest for learning generalizable node representations in a self-supervised manner. In general, the contrastive learning process in GCL is performed on top of the representations learned by a graph neural network (GNN) backbone, which transforms and propagates the node contextual information based on its local neighborhoods. However, nodes sharing similar characteristics may not always be closely connected, which poses a great challenge for unsupervised GCL efforts due to their inherent limitations in capturing such global graph knowledge. In this work, we address their inherent limitations by proposing a simple yet effective framework – *Simple Neural Networks with Structural and Semantic Contrastive Learning* (S^3-CL). Notably, by virtue of the proposed structural and semantic contrastive learning algorithms, even a simple neural network can learn expressive node representations that preserve valuable global structural and semantic patterns. Our experiments demonstrate that the node representations learned by S^3-CL achieve superior performance on different downstream tasks compared with the state-of-the-art unsupervised GCL methods. Implementation and more experimental details are publicly available at https://github.com/kaize0409/S-3-CL.

1 Introduction

Learning expressive node representations of graph-structured data plays an essential role in a variety of real-world applications, ranging from social network analysis (Kipf and Welling 2017), to drug discovery (Fout et al. 2017), to financial fraud detection (Ding et al. 2019). Recently, graph neural networks (GNNs), which generally follow a recursive message-passing scheme, have emerged as powerful architectures in graph machine learning (Kipf and Welling 2017; Veličković et al. 2018; Hamilton, Ying, and Leskovec 2017; Wu et al. 2019; Ding et al. 2020; Wang et al. 2020). Though GNNs are empirically effective in handling supervised or semi-supervised graph machine learning tasks, the labor-intensive and resource-expensive data labeling cost is meanwhile unbearable (Ding et al. 2022b; Zhang et al. 2022; Ding et al. 2022c). To relieve the burdensome reliance on human-annotated labels, unsupervised (self-supervised) node representation learning with GNNs has drawn much research attention lately (Kipf and Welling 2016; Veličković et al. 2019; You et al. 2020).

More recently, contrastive learning (He et al. 2020; Chen et al. 2020b) has been actively explored to advance the performance of GNNs in graph self-supervised learning (Veličković et al. 2019; You et al. 2020; Hassani and Khasahmadi 2020; Qiu et al. 2020; Zhu et al. 2020b). In general, graph contrastive learning (GCL) methods learn representations by creating two augmented views of each graph element and maximizing the agreement between the encoded representations of the two augmented views. Correspondingly, the relevant view pairs (positive) will be pulled together, and the irrelevant view pairs (negative) will be pushed away in the latent space. With only non-semantic labels, unsupervised GCL can provide generalizable node representations for various downstream tasks (You et al. 2020; Hassani and Khasahmadi 2020; Du et al. 2021), becoming a prevailing paradigm in unsupervised node representation learning.

Despite the success, the research of unsupervised GCL is still in its infancy – most of the existing GCL methods learn node representations based on the information from the local neighborhoods due to the shallow property of conventional GNNs. While for real-world graphs, *nodes sharing similar characteristics may not always be closely connected*, requiring the learning algorithm to retain such "global" awareness. However, it is a non-trivial task for the existing GCL methods built on top of shallow GNNs since they have inherent limitations in capturing either *structural global knowledge* or *semantic global knowledge*. Specifically: **(i)** from the structural perspective, long-range node interactions are highly desired for capturing structural global knowledge, especially for many downstream tasks that have large problem radii (Alon and Yahav 2021). To this end, a straightforward way is to employ a deeper GNN encoder to encode the augmented graphs. However, directly stacking multiple GNN layers will not only lead to information distortion caused by the oversmoothing issue (Chen et al. 2020a), but also introduce additional training parameters that hamper the model training efficiency; and **(ii)** from the semantic perspective, existing unsupervised GCL methods predominately focus on instance-level contrast that leads to a latent space where all nodes are well-separated and each node is locally smooth (Li et al. 2021) (i.e., input with different augmentations have similar representations), while the underlying semantic structure

*Indicates equal contribution.

(i.e., intra-cluster compactness and inter-cluster separability) of the input graph is largely ignored (Li et al. 2021). The lack of prior knowledge of ground-truth labels (e.g., cluster/class numbers) leaves a significant gap for unsupervised GCL to consolidate the semantic structure from a global view in the latent space. Yet, how to bridge this gap remains unattended.

In this paper, we address the aforementioned limitations by proposing a simple yet effective GCL framework, namely, S^3-CL (*Simple Neural Networks with Structural and Semantic Contrastive Learning*). The proposed two new contrastive learning algorithms enable the framework to outperform other GCL counterparts with a much simpler and parameter-less encoding backbone, such as an MLP or even a one-layer neural network. To capture long-range node interactions without oversmoothing, the *structural contrastive learning* algorithm first generates multiple augmented views of the input graph based on different feature propagation scales (i.e., multi-scale feature propagation). Then by performing contrastive learning on the node representations learned from the local and multiple high-order views, the encoder network can improve node-wise discrimination by exploiting the consistency between the local and global structure information of each node. In the meantime, the *semantic contrastive learning* algorithm further enhances intra-cluster compactness and inter-cluster separability to better consolidate the semantic structure from a global view. Specifically, it infers the clusters among nodes and their corresponding prototypes by a new Bayesian nonparametric algorithm and then performs semantic contrastive learning to enforce those nodes that are semantically similar to cluster around their corresponding cluster prototypes in the latent space. By jointly optimizing the structural and semantic contrastive losses, the pre-trained encoder network can learn highly expressive node representations for various downstream tasks without using any human-annotated labels. We summarize our contributions as follows:

- We develop a new GCL framework S^3-CL, which can learn expressive node representations in a self-supervised fashion by using a simple and parameter-less encoding backbone.
- We propose structural and semantic contrastive learning algorithms, which can be used for explicitly capturing the global structural and semantic patterns of the input graph.
- We conduct extensive experiments to show that our approach significantly outperforms the state-of-the-art GCL counterparts on various downstream tasks.

2 Preliminaries

We start by introducing the notations used throughout the paper. An attributed graph with N nodes can be formally represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the set of nodes and edges respectively. Let $\mathbf{A} \in \{0,1\}^{N \times N}$ be the adjacency matrix of graph $\mathcal{G}$. $\mathbf{A}_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. $\tilde{\mathbf{A}}$ stands for the adjacency matrix for a graph with added self-loops $\mathbf{I}$. We let $\mathbf{D}$ and $\tilde{\mathbf{D}}$ denote the diagonal degree matrix of $\mathbf{A}$ and $\tilde{\mathbf{A}}$ respectively. $\mathbf{x}_i$ is the i-th row of the attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, which denotes the feature of node v_i. Hence, an attributed graph can also be described as $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ for simplicity.

Graph Contrastive Learning. In general, graph contrastive learning aims to pre-train a graph encoder that can maximize the node-wise agreement between two augmented views of the same graph element in the latent space via a contrastive loss. Generally, given an attributed graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, two different augmented views of the graph, denoted as $\mathcal{G}^{(1)} = (\mathbf{X}^{(1)}, \mathbf{A}^{(1)})$ and $\mathcal{G}^{(2)} = (\mathbf{X}^{(2)}, \mathbf{A}^{(2)})$, are generated through the data augmentation function(s). The node representations on $\mathcal{G}^1$ and $\mathcal{G}^{(2)}$ are denoted as $\mathbf{H}^{(1)} = f_{\boldsymbol{\theta}}(\mathbf{X}^{(1)}, \mathbf{A}^{(1)})$ and $\mathbf{H}^{(2)} = f_{\boldsymbol{\theta}}(\mathbf{X}^{(2)}, \mathbf{A}^{(2)})$, where $f_{\boldsymbol{\theta}}(\cdot)$ is an encoder network. The agreement between the node representations is commonly measured through Mutual Information (MI). Thus, the contrastive objective can be generally formulated as:

$$\max_{\theta} \sum_{i=1}^{N} \mathcal{MI}(\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)}). \tag{1}$$

Following this formulation, Deep Graph Infomax (DGI) (Veličković et al. 2019) is the first method that contrasts the patch representations with high-level graph representations by maximizing their mutual information. MVGRL (Hassani and Khasahmadi 2020) adopts graph diffusion to generate an augmented view, and contrast representations of first-order neighbors with a graph diffusion. GCC (Qiu et al. 2020) and GRACE (Zhu et al. 2020a) create the augmented views by sampling subgraphs. MERIT (Jin et al. 2021) adopts a siamese self-distillation network and performs contrastive learning across views and networks at the same time. Nonetheless, existing unsupervised GCL methods only focus on short-range node interactions and are also ineffective in capturing the semantic structure of graphs.

3 Methodology

In this paper, we propose a novel graph contrastive learning framework S^3-CL for unsupervised/self-supervised node representation learning. The overall framework is illustrated in Figure 1. Our proposed framework consists of three main components: (i) a simple (e.g., 1-layer) encoder network; (ii) a structural contrastive learning algorithm; and (iii) a semantic contrastive learning algorithm.

3.1 Structural Contrastive Learning

Existing GCL methods for unsupervised node representation learning aim to achieve node-wise discrimination by maximizing the agreement between the representations of the same graph element in different augmented views. Despite their success, they commonly ignore the global structure knowledge due to the limitations of either the adopted data augmentation function or the GNN encoder. In this work, we propose the *structural contrastive learning* algorithm, which enables a simple neural network to capture both local and global structural knowledge by performing contrastive learning on multi-scale augmented graph views.

Multi-scale Feature Propagation. In order to capture long-range node interactions without suffering the oversmoothing issue, in our structural contrastive learning algorithm, we propose to adopt *multi-scale feature propagation* to augment the input graph from the structural perspective. Compared to arbitrarily modifying the graph structure such as perturbing

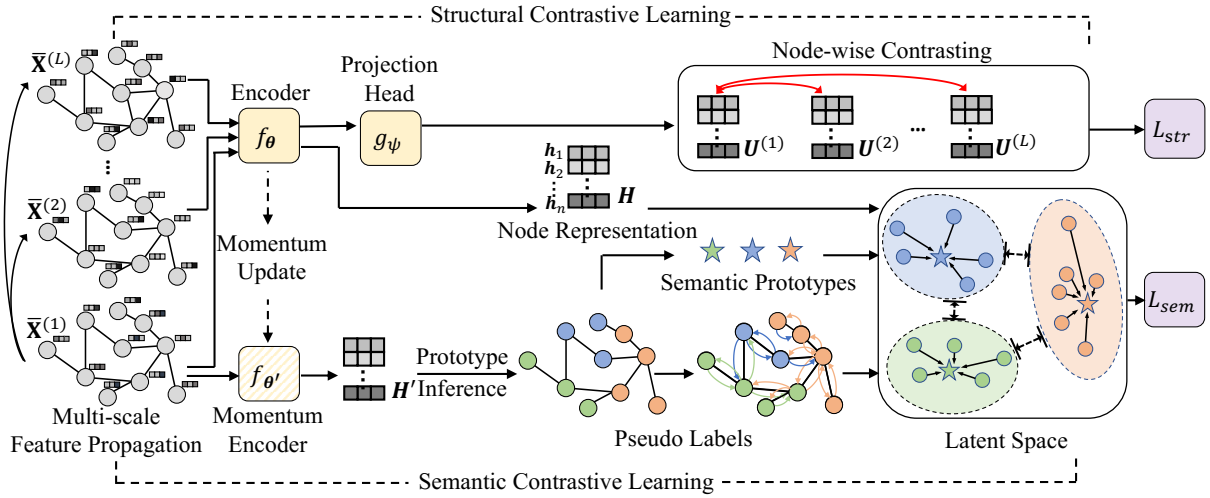


Figure 1: Illustration of the overall framework S^3-CL for self-supervised node representation learning.

edges or nodes, feature propagation not only allows incorporating long-range node interactions but also mitigates the noises in the original graph (Ding et al. 2022b). Unlike existing GCL algorithms that perform only two augmentations for each instance, we perform feature propagation with different scales to generate L augmented feature matrices $\{\bar{\mathbf{X}}^{(l)}\}_{l=1}^{L}$, each of which encodes the l-hop node interactions in the graph. Then each augmented feature matrix $\bar{\mathbf{X}}^{(l)}$ can be encoded by a encoder network $f_{\boldsymbol{\theta}}(\cdot)$ and the corresponding node representations can be computed by:

$$\mathbf{H}^{(l)} = f_{\boldsymbol{\theta}}(\bar{\mathbf{X}}^{(l)}) = \mathrm{ReLU}(\bar{\mathbf{X}}^{(l)}\boldsymbol{\Theta}), \quad \bar{\mathbf{X}}^{(l)} = \mathbf{T}^{l}\mathbf{X}, \quad (2)$$

where $\mathbf{T} \in \mathbb{R}^{N\times N}$ is a generalized transition matrix and we take $\mathbf{T} = \tilde{\mathbf{A}}_{sym} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ in this work. $\mathbf{H}^{(1)}$ is learned from a local view as the message-passing is only enabled between direct neighbors, while $\{\mathbf{H}^{(l)}\}_{l=2}^{L}$ are learned from a set of high-order views that encode the long-range node interactions at different scales.

It is noteworthy that our model inherently separates the feature propagation step, i.e., $\bar{\mathbf{X}}^{(l)} = \mathbf{T}^{l}\mathbf{X}$, and transformation step, i.e., $f_{\boldsymbol{\theta}}(\bar{\mathbf{X}}^{(l)})$ into the data augmentation and representation learning modules, respectively. Compared to standard GNNs that couple the two steps together in each layer, this decoupling strategy allows the model to perform the high-order feature propagation without conducting non-linear transformations, reducing the risk of over-smoothing (Feng et al. 2020; Ding et al. 2022a) in contrastive learning. In the meantime, we can use a much simpler encoding backbone to transform the augmented features to node representations without stacking multiple GNN layers.

Structural Contrastive Objective. In our structural contrastive learning, we aim to maximize the agreement between the representations of each node learned from the local view and its different high-order views by maximizing their mutual information. Instead of directly contrasting the output of the encoder network, we follow previous research in contrastive learning (Chen et al. 2020b) and apply a *projection head* $g_{\boldsymbol{\psi}}(\cdot)$ to the node representations computed by the encoder network. As such, the representations we contrast in our structural contrastive learning can be denoted by $\{\mathbf{U}^{(l)}\}_{l=1}^{L}$, where $\mathbf{U}^{(l)} = g_{\boldsymbol{\psi}}(\mathbf{H}^{(l)})$, and $g_{\boldsymbol{\psi}}(\cdot)$ is a two-layer MLP in our implementation.

In our work, we adopt InfoNCE (Oord, Li, and Vinyals 2018) to estimate the lower bound of the mutual information between the node representations learned from a local view $\mathbf{U}^{(1)}$ and different high-order views $\{\mathbf{U}^{(l)}\}_{l=2}^{L}$ of the input graph. The loss function of structural contrastive learning can be defined as:

$$\mathcal{L}_{str} = -\sum_{i=1}^{N}\sum_{l=2}^{L}\log\frac{\exp(\mathbf{u}_i^{(1)}\cdot\mathbf{u}_i^{(l)}/\tau_1)}{\sum_{j=1}^{M+L-1}\exp(\mathbf{u}_i^{(1)}\cdot\mathbf{u}_j^{(l)}/\tau_1)}, \quad (3)$$

where τ_1 is the temperature parameter. Note that $\{\mathbf{u}_j^{(l)}\}_{j=1}^{M+L-1}$ contains $L-1$ positive examples and M negative examples sampled from augmented views of other nodes.

By performing the proposed structural contrastive learning based on multi-scale augmentations of the input graph, the encoder network $f_{\boldsymbol{\theta}}(\cdot)$ not only encourages accurate node-wise discrimination but also captures multi-scale global structural knowledge during the learning process. The resulted node representations $\mathbf{H}$ can be computed by feeding the mixed-order propagated features $\bar{\mathbf{X}}$ to the encoder network as:

$$\mathbf{H} = f_{\boldsymbol{\theta}}(\bar{\mathbf{X}}) = \mathrm{ReLU}(\bar{\mathbf{X}}\boldsymbol{\Theta}), \quad \bar{\mathbf{X}} = \frac{1}{L}\sum_{l=0}^{L}\mathbf{T}^{l}\mathbf{X}. \quad (4)$$

This enables the learned node representations to preserve both local and global structure information compared with directly using $\mathbf{T}^{L}\mathbf{X}$ (Xu et al. 2018; Feng et al. 2020).

3.2 Semantic Contrastive Learning

Despite the structural contrastive learning algorithm can provide better node-wise discrimination by exploiting the global structural knowledge based on the multi-scale propagated

features, it has the same limitation as existing GCL efforts – cannot explicitly encode the semantic structure of the input graph. To further capture the semantic global knowledge, we propose a *semantic contrastive learning* algorithm that encourages the intra-cluster compactness and inter-cluster separability in the semantic latent space.

Since the prior knowledge of node clusters is unknown, we propose to iteratively infer the clusters among nodes and the corresponding prototypes based on the learned node representations, and perform semantic contrastive learning to promote those nodes that are semantically similar clustering around their corresponding cluster prototypes.

We denote the cluster prototype representation via a matrix $\mathbf{C} \in \mathbb{R}^{K \times D'}$, where K is the number of prototypes inferred from the data. We use $\mathbf{c}_k$ to denote the k-th row of $\mathbf{C}$, which is the representation of the k-th prototype in the latent space. The prototype assignments or pseudo labels of nodes are denoted by $\mathcal{Z} = \{z_i\}_{i=1}^{n}$, where $z_i \in \{1, ..., K\}$ is the pseudo label of node v_i.

Bayesian Non-parametric Prototype Inference. A key function of our semantic contrastive learning algorithm is to infer highly representative cluster prototypes. However, the optimal number of clusters is unknown under the setting of unsupervised node representation learning. To bridge the gap, we propose a Bayesian non-parametric prototype inference algorithm to approximate the optimal number of clusters and simultaneously compute the cluster prototypes. Specifically, we build a Dirichlet Process Mixture Model (DPMM) and assume the distribution of node representations is a mixture of Gaussians, in which each component is used to model the prototype of a cluster. Note that the components share the same fixed covariance matrix $\sigma\mathbf{I}$. The DPMM model is defined as:

$$G \sim \text{DP}(G_0, \alpha), \quad \boldsymbol{\phi}_i \sim G, \quad \mathbf{h}_i \sim \mathcal{N}(\boldsymbol{\phi}_i, \sigma\mathbf{I}), \quad (5)$$

where G is a Gaussian distribution drawn from the Dirichlet process $\text{DP}(G_0, \alpha)$, and α is the concentration parameter for $\text{DP}(G_0, \alpha)$. $\boldsymbol{\phi}_i$ is the mean of the Gaussian sampled for node representation $\mathbf{h}_i$. G_0 is the prior over means of the Gaussians. We take G_0 to be a zero-mean Gaussian $\mathcal{N}(\mathbf{0}, \rho\mathbf{I})$, where $\rho\mathbf{I}$ is the covariance matrix.

Next, we use a collapsed Gibbs sampler (Resnik and Hardisty 2010) to infer the Gaussian components. The Gibbs sampler iteratively samples pseudo labels for the nodes given the means of the Gaussian components and samples the means of the Gaussian components given the pseudo labels of the nodes. Following (Kulis and Jordan 2011), such a process is almost equivalent to K-Means when the variance of the Gaussian components $\sigma \to 0$. The almost zero variance eliminates the need to estimate the variance σ, thus making the inference efficient. Let $\tilde{K}$ denote the number of inferred prototypes at the current iteration step, the prototype assignment update can be formulated as:

$$\begin{aligned} z_i &= \arg\min_k \{d_{ik}\}, \\ d_{ik} &= \begin{cases} ||\mathbf{h}_i - \mathbf{c}_k||^2 & \text{for } k = 1, ..., \tilde{K} \\ \xi & \text{for } k = \tilde{K} + 1, \end{cases} \end{aligned} \quad (6)$$

where d_{ik} is the distance to determine the pseudo labels of node representation $\mathbf{h}_i$. ξ is the margin to initialize a new prototype. With the formulation in Equation (6), a node will be assigned to the prototype modeled by the Gaussian component corresponding to the closest mean of Gaussian, unless the squared Euclidean distance to the closest mean is greater than ξ. In this case, we initialize a new prototype with such node representation. After obtaining the pseudo labels, the cluster prototype representations can be computed by: $\mathbf{c}_k = \sum_{z_i=k} \mathbf{h}_i / \sum_{z_i=k} 1$, for $k = 1, ..., \tilde{K}$.

Note that we iteratively update prototype assignments and prototype representations till convergence, and we set the number of prototypes K to be the number of inferred prototypes $\tilde{K}$. Afterward, we refine the cluster prototypes using label propagation, and the algorithm details can be found in the Appendix of our extended version[1] due to the space limit.

Semantic Contrastive Objective. After obtaining the prototype assignments $\mathcal{Z}$ and prototype representations $\mathbf{C}$, our semantic contrastive objective aims to consolidate the semantic structure (i.e., intra-cluster compactness and inter-cluster separability) of the learned node representation $\mathbf{H}$ by updating the encoder parameter $\boldsymbol{\theta}$. To this end, we maximize the likelihood of each node in the graph given $\boldsymbol{\theta}$ and $\mathbf{C}$:

$$\begin{aligned} Q(\boldsymbol{\theta}) &= \sum_{n=1}^{N} \log p(\mathbf{h}_i | \boldsymbol{\theta}, \mathbf{C}) \\ &= \sum_{n=1}^{N} \log \sum_{k=1}^{K} p(\mathbf{h}_i, k | \boldsymbol{\theta}, \mathbf{C}), \end{aligned} \quad (7)$$

where p is the probability density function. Directly optimizing log-likelihood $Q(\boldsymbol{\theta})$ is intractable as the labels of nodes are unknown. Instead, we optimize the variational lower bound of $Q(\boldsymbol{\theta})$, given by:

$$\begin{aligned} Q(\boldsymbol{\theta}) \geq & \sum_{i=1}^{N} \sum_{k=1}^{K} p(k|\mathbf{h}_i) \log \frac{p(\mathbf{h}_i, k | \boldsymbol{\theta}, \mathbf{C})}{p(k|\mathbf{h}_i)} \\ = & \sum_{i=1}^{N} \sum_{k=1}^{K} p(k|\mathbf{h}_i) \log p(\mathbf{h}_i, k | \boldsymbol{\theta}, \mathbf{C}) \\ & - \sum_{i=1}^{N} \sum_{k=1}^{K} p(k|\mathbf{h}_i) \log p(k|\mathbf{h}_i). \end{aligned} \quad (8)$$

Note that we can drop the second term of the right-hand side of Equation (8) as it is a constant. To maximize the remaining part, we can estimate $p(k|\mathbf{h}_i)$ by $p(k|\mathbf{h}_i, \boldsymbol{\theta}, \mathbf{C}) = \mathbb{1}_{\{k=z_i\}}$, as we assign $\mathbf{h}_i$ to cluster z_i given $\mathbf{C}$ in our DPMM model. Thus, we can maximize $Q(\boldsymbol{\theta})$ by minimizing the following loss function:

$$\mathcal{L}_{sem} = -\sum_{i=1}^{N} \log p(\mathbf{h}_i, z_i | \boldsymbol{\theta}, \mathbf{C}). \quad (9)$$

Under the assumption of a uniform prior distribution of node representation, we have $p(\mathbf{h}_i, z_i | \boldsymbol{\theta}, \mathbf{C}) \propto p(z_i | \mathbf{h}_i, \boldsymbol{\theta}, \mathbf{C})$. Since the distribution of node representation around each prototype generated by the DPMM is an isotropic Gaussian, we can estimate $p(k|\mathbf{h}_i, \boldsymbol{\theta}, \mathbf{C})$ by

[1] https://arxiv.org/abs/2202.08480

$\exp(||\mathbf{h}_i - \mathbf{c}_{z_i}||^2/\sigma^2)/\sum_{i=1}^{K}\exp(||\mathbf{h}_i - \mathbf{c}_k||^2/\sigma^2)$. After applying ℓ_2 normalization on the representation of nodes and prototypes, we can estimate $p(z_i|\mathbf{h}_i, \boldsymbol{\theta}, \mathbf{C})$ by:

$$p(z_i|\mathbf{h}_i, \boldsymbol{\theta}, \mathbf{C}) = \frac{\exp(\mathbf{h}_i \cdot \mathbf{c}_{z_i}/\tau_2)}{\sum_{k=1}^{K}\exp(\mathbf{h}_i \cdot \mathbf{c}_k/\tau_2)}, \tag{10}$$

where $\mathbf{c}_{z_i}$ is the representations of z_i-th prototype. The temperature parameter $\tau_2 \propto \sigma^2$ is related to the concentration of node representation around each prototype, and σ is the variance of the Gaussians in the DPMM model defined by Equation (5). For the simplicity of training, we directly take τ_2 as a hyperparameter. Taking Equation (10) into Equation (9), we can maximize $Q(\boldsymbol{\theta})$ by minimizing the following loss function:

$$\mathcal{L}_{sem} = -\sum_{i=1}^{N}\log\frac{\exp(\mathbf{h}_i \cdot \mathbf{c}_{z_i}/\tau_2)}{\sum_{k=1}^{K}\exp(\mathbf{h}_i \cdot \mathbf{c}_k/\tau_2)}. \tag{11}$$

3.3 Model Learning

Given the proposed S^3-CL learning framework, our goal is to learn expressive node representations that preserve both valuable structural and semantic knowledge without any semantic labels. In this section, we will introduce the overall loss function, and also the optimization of the proposed framework with regard to the network parameters, prototype assignments, and prototype representations.

Overall Loss. To train our model in an end-to-end fashion and learn the encoder $f_{\boldsymbol{\theta}}(\cdot)$, we jointly optimize both the structural and semantic contrastive learning losses. The overall objective function is defined as:

$$\mathcal{L} = \gamma\mathcal{L}_{str} + (1-\gamma)\mathcal{L}_{sem}, \tag{12}$$

where we aim to minimize $\mathcal{L}$ during training, and γ is a balancing parameter to control the contribution of each contrastive learning loss. For the sake of the stability of the training of the encoder, we apply our Bayesian non-parametric prototype inference algorithm on the node representations computed by a momentum encoder (He et al. 2020).

Notably, in semantic contrastive learning, the computed pseudo labels $\mathcal{Z}$ can be utilized in the negative example sampling process in our structural contrastive learning to avoid sampling bias issues (Chuang et al. 2020). We select negative samples in Equation (3) for each node from nodes assigned to different prototypes.

Model Optimization via EM. Specifically, we adopt EM algorithm to alternately estimate the posterior distribution $p(z_i|\mathbf{x}_i, \boldsymbol{\theta}, \mathbf{C})$ and optimize the network parameters $\boldsymbol{\theta}$. We describe the details for the E-step and M-step applied in our methods as follows:

- **E-step.** In this step, we fix the network parameter $\boldsymbol{\theta}$, and estimate the prototypes $\mathbf{C}$ and the prototype assignment $\mathcal{Z}$ with our proposed Bayesian non-parametric prototype inference algorithm.
- **M-step.** Given the posterior distribution computed by the E-step, we aim to maximize the expectation of log-likelihood $Q(\boldsymbol{\theta})$, by directly optimizing the semantic contrastive loss function $\mathcal{L}_{sem}$. In order to perform structural and semantic contrastive learning at the same time, we instead optimize a joint overall loss function as formulated in Equation (12).

Algorithm 1: The learning algorithm of S^3-CL.

Input: Attribute matrix $\mathbf{X}$; adjacency matrix $\mathbf{A}$; propagation step L
Output: Pretrained encoder network $f_{\boldsymbol{\theta}}(\cdot)$

```
Initialize encoder parameter θ and θ'
while not converge do
    Compute node representations of different augmented views {H^(l)}_{l=1}^L and {U^(l)}_{l=1}^L
    Compute the prototype representations C and prototype assignments Z        ▷ E-step Update
    Calculate loss L_str and L_sem by Equation (3) and Equation (11), respectively
    L = γ L_str + (1 − γ) L_sem
    Update θ by minimizing L                                                   ▷ M-step Update
    Update momentum encoder θ'
return the encoder network f_θ(·)
```

Algorithm 1 outlines the learning process of the proposed framework. After the self-supervised pre-training is done, the pre-trained encoder can be directly used to generate node representations for various downstream tasks.

4 Experiments

4.1 Experimental Settings

Evaluation Datasets. In our experiments, we evaluate S^3-CL on six public benchmark datasets that are widely used for node representation learning, including Cora (Sen et al. 2008), Citeseer (Sen et al. 2008), Pubmed (Namata et al. 2012), Amazon-P (Shchur et al. 2018), Coauthor CS (Shchur et al. 2018) and ogbn-arxiv (Hu et al. 2020). Cora, Citeseer, and Pubmed are the three most widely used citation networks. Amazon-P is a co-purchase graph and Coauthor CS is a co-authorship graph. The ogbn-arxiv is a large-scale citation graph benchmark dataset.

Compared Methods. To demonstrate the effectiveness of our proposed method, six state-of-the-art graph self-supervised learning methods are compared in our experiments, including DGI (Veličković et al. 2019), MVGRL (Hassani and Khasahmadi 2020), GMI (Peng et al. 2020), GRACE (Zhu et al. 2020a), MERIT (Jin et al. 2021), and SUGRL (Mo et al. 2022). As we consider node classification as our downstream task, we also include five representative supervised node classification methods, namely MLP (Veličković et al. 2019), LP (Zhu, Ghahramani, and Lafferty 2003), GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), and SGC (Wu et al. 2019), as baselines for the evaluation on the node classification task. To evaluate the model performance for node clustering, we compare S^3-CL against methods including K-Means (Lloyd 1982), GAE (Kipf and Welling 2016), adversarially regularized GAE (ARGA) and VGAE (ARVGA) (Pan et al. 2018), GALA (Park et al. 2019), DGI, DBGAN (Zheng et al. 2020), MVGRL, MERIT, and SUGRL.

4.2 Evaluation Results

Node Classification. To evaluate the trained encoder network, we adopt a linear evaluation protocol by training a separate logistic regression classifier on top of the learned

Methods	Cora	Citeseer	Pubmed	Amazon-P	Coauthor CS	ogbn-arxiv
SUPERVISED						
MLP	55.2 ± 0.4	46.5 ± 0.5	71.4 ± 0.3	78.5 ± 0.2	76.5 ± 0.3	55.5 ± 0.2
LP (Zhu, Ghahramani, and Lafferty 2003)	68.0 ± 0.5	45.3 ± 0.6	63.0 ± 0.3	75.4 ± 0.0	74.3 ± 0.0	68.3 ± 0.0
GCN (Kipf and Welling 2017)	81.7 ± 0.4	70.5 ± 0.3	79.4 ± 0.4	87.3 ± 1.0	91.8 ± 0.1	71.7 ± 0.3
GAT (Veličković et al. 2018)	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	86.2 ± 1.5	90.5 ± 0.7	$\mathbf{73.2 \pm 0.2}$
SGC (Wu et al. 2019)	81.5 ± 0.2	73.1 ± 0.1	79.7 ± 0.4	88.3 ± 1.1	91.5 ± 0.3	69.8 ± 0.2
SELF-SUPERVISED + FINE-TUNING						
DGI (Veličković et al. 2019)	81.7 ± 0.6	71.5 ± 0.7	77.3 ± 0.6	83.1 ± 0.3	90.0 ± 0.3	67.1 ± 0.4
GMI (Peng et al. 2020)	82.7 ± 0.2	73.0 ± 0.3	80.1 ± 0.2	85.1 ± 0.0	91.0 ± 0.0	69.6 ± 0.3
MVGRL (Hassani and Khasahmadi 2020)	82.9 ± 0.7	72.6 ± 0.7	79.4 ± 0.3	87.3 ± 0.1	91.3 ± 0.1	71.3 ± 0.2
GRACE (Zhu et al. 2020a)	80.0 ± 0.4	71.7 ± 0.6	79.5 ± 1.1	81.8 ± 0.8	90.1 ± 0.8	71.1 ± 0.2
MERIT (Jin et al. 2021)	83.1 ± 0.6	$\underline{74.0 \pm 0.7}$	80.1 ± 0.4	$\underline{88.8 \pm 0.4}$	$\underline{92.4 \pm 0.4}$	71.7 ± 0.1
SUGRL (Mo et al. 2022)	$\underline{83.4 \pm 0.5}$	73.0 ± 0.5	$\mathbf{81.9 \pm 0.5}$	88.5 ± 0.2	92.2 ± 0.5	69.3 ± 0.2
S^3-CL (ours)	$\mathbf{84.5 \pm 0.4}$	$\mathbf{74.6 \pm 0.4}$	$\underline{80.8 \pm 0.3}$	$\mathbf{89.0 \pm 0.5}$	$\mathbf{93.1 \pm 0.4}$	$\underline{72.8 \pm 0.3}$

Table 1: Node classification performance comparison on benchmark datasets.

Methods	Cora			Citeseer			Pubmed		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-Means	49.2	32.1	22.9	54.0	30.5	27.8	59.5	31.5	28.1
GAE (Kipf and Welling 2016)	59.6	42.9	34.7	40.8	17.6	12.4	67.2	27.7	27.9
ARGA (Pan et al. 2018)	64.0	44.9	35.2	57.3	35.0	34.1	66.8	30.5	29.5
ARVGA (Pan et al. 2018)	64.0	45.0	37.4	54.4	26.1	24.5	69.0	29.0	30.6
GALA (Park et al. 2019)	74.5	57.6	53.1	69.3	44.1	44.6	69.3	32.7	32.1
DGI (Veličković et al. 2019)	55.4	41.1	32.7	51.4	31.5	32.6	58.9	27.7	31.5
DBGAN (Zheng et al. 2020)	74.8	56.0	54.0	67.0	40.7	41.4	69.4	32.4	32.7
MVGRL (Hassani and Khasahmadi 2020)	73.2	56.2	51.9	68.1	43.2	43.4	69.3	34.4	32.3
MERIT (Jin et al. 2021)	73.6	57.1	52.8	68.9	43.9	44.1	69.5	34.7	32.8
SUGRL (Mo et al. 2022)	73.9	58.5	53.0	70.5	45.8	47.0	69.5	35.0	33.4
S^3-CL (ours)	**75.1**	**60.7**	**56.6**	**71.2**	**46.3**	**48.5**	**71.3**	**36.0**	**34.7**

Table 2: Node clustering performance comparison on benchmark datasets.

node representations. We follow the evaluation protocols in previous works (Veličković et al. 2019; Hu et al. 2020) for node classification. The mean classification accuracy with standard deviation on the test nodes after 10 runs of training is reported. To avoid the out-of-memory issue when evaluating MVGRL, GRACE, and MERIT on the ogbn-arxiv dataset, we subsample 512 nodes as negative samples for each node during the self-supervised learning phase.

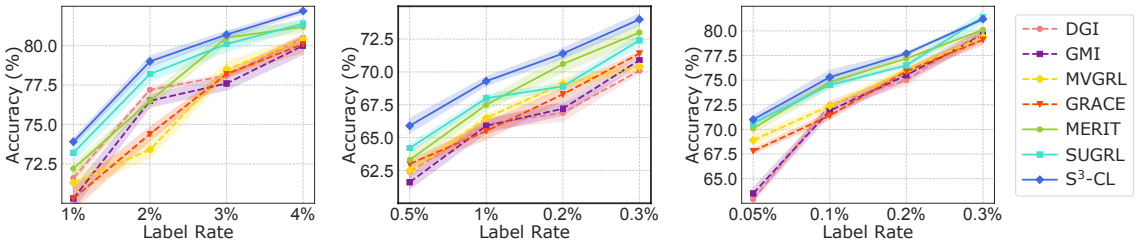
The node classification results of different methods are reported in Table 1. We can clearly see that S^3-CL outperforms the state-of-the-art self-supervised node representation learning methods across the five public benchmarks. Such superiority mainly stems from two factors: (i) our approach S^3-CL grants each node access to information of nodes in a larger neighborhood; (ii) S^3-CL infers the semantic information of nodes, and enforces intra-cluster compactness and inter-cluster separability on the node representation. With the help of this extra information, node representations generated by S^3-CL are more informative and distinctive. Without access to labels, S^3-CL even outperforms supervised methods like SGC and GAT.

Node Clustering. To evaluate the quality of the node representations learned by different methods, we conduct experiments on node clustering. We follow the same evaluation protocol as in (Hassani and Khasahmadi 2020). K-Means is applied on the learned node representation to get clustering results. We use accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) to measure the performance of clustering. We report the averaged clustering results over 10 times of execution.

The clustering results are displayed in Table 2. It is observed that our approach achieves remarkable performance gain over compared methods. For example, the NMI on Cora is improved by 2.2% against the previous SOTA method SUGRL. Such improvement greatly attributes to the fact that S^3-CL explores the semantic information of nodes instead of enforcing node-wise discrimination alone as other GCL methods. Thus, the node representation learned by S^3-CL works well for clustering algorithms.

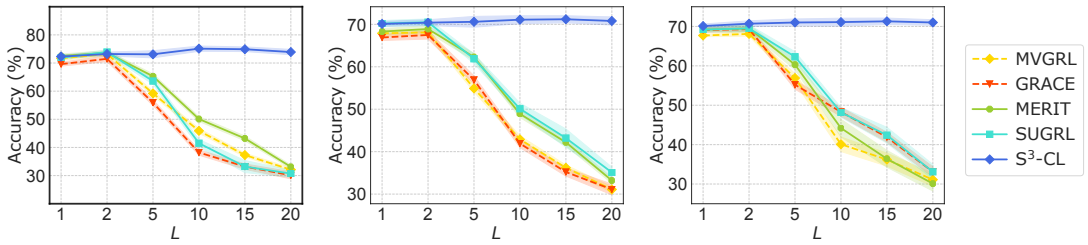
Node Classification with Few Labels. We further evaluate the impact of label rate on the downstream node classification task. Specifically, we evaluate all self-supervised learning methods from Table 1 under different low-resource settings. The results in Figure 2 show that our proposed framework S^3-CL can still outperform existing methods given a lower label rate. It validates that the node representations learned by our approach S^3-CL can encode valuable structural and semantic knowledge from the input graph. As a result, the node representations can be effectively used for the node classification task even with an extremely small label ratio.

Effect of Feature Propagation. Next, we investigate the effect of multi-scale feature propagation in the structural



(a) Cora (b) Citeseer (c) Pubmed

Figure 2: Node classification results with limited training labels.



(a) Cora (b) Citeseer (c) Pubmed

Figure 3: Node clustering results of GCL methods with various propagation steps (L).

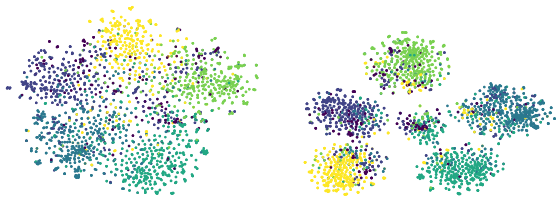
contrastive learning by altering the propagation steps L. A larger L allows message-passing within a larger neighborhood for learning the node representations. To demonstrate the power of our approach in utilizing structural global knowledge, we compare S^3-CL against GRACE, MVGRL, MERIT, and SUGRL with different numbers of layers L. The node clustering accuracy of different methods is shown in Figure 3. By increasing the propagation steps (number of layers), we can clearly observe that existing unsupervised GCL methods severely degrade due to the oversmoothing issue. In contrast, S^3-CL consistently achieves improved performance by making use of information in a larger neighborhood for node representation learning.

Ablation Study. To validate the effectiveness of the structural contrastive learning and semantic contrastive learning in S^3-CL, we conduct an ablation study on Citesser, Cora, and Pubmed with two variants of S^3-CL, each of which has one of the contrastive learning components removed. The node classification results are shown in Table 3. We can observe that the performance of S^3-CL degrades when any of the components are removed. Our S^3-CL using all components achieves the best performance as the structural and semantic contrastive components complement each other. Hence, the effectiveness of each component is verified.

Method	Citeseer	Cora	Pubmed
w/o structural	73.1±0.2	83.3±0.3	80.0± 0.3
w/o semantic	71.9±0.4	82.2±0.5	79.3± 0.2
S^3-CL	**74.6±0.4**	**84.5±0.4**	**80.8±0.3**

Table 3: Ablation study on contrastive components.

Representation Visualization. To visually show the superior quality of the node representations learned by S^3-CL, we use t-SNE to visualize and compare the learned node representations between S^3-CL and the best-performing baseline on Citeseer, i.e., SUGRL. The visualization results are shown in Figure 4, where each dot represents the representation of a node, and the color of the dot denotes its ground-truth label. From the figure, we can observe that though some classes can be identified by SUGRL, the boundaries between different classes are unclear. Our proposed model is able to enforce better intra-cluster compactness and inter-cluster separability.



(a) SUGRL (b) S^3-CL

Figure 4: Representation visualization on the Citeseer dataset.

5 Conclusion

In this paper, we propose a new GCL framework named S^3-CL, which can effectively capture the global knowledge from both structural and semantic perspectives for unsupervised node representation learning. By jointly optimizing the structural and semantic contrastive learning losses, we can build the encoder network with simple neural networks to learn expressive node representations for different downstream tasks without using any human-annotated labels. We conduct extensive experiments and demonstrate that S^3-CL can outperform the state-of-the-art unsupervised GCL counterparts on multiple benchmark graph datasets.

Acknowledgments

This work is supported by NSF (No. 2229461), ARO (No. W911NF2110088), ONR (No. N00014-21-1-4002), and ARL (No. W911NF2020124).

References

Alon, U.; and Yahav, E. 2021. On the bottleneck of graph neural networks and its practical implications. In *ICLR*.

Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; and Sun, X. 2020a. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*.

Chuang, C.-Y.; Robinson, J.; Yen-Chen, L.; Torralba, A.; and Jegelka, S. 2020. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *SDM*.

Ding, K.; Wang, J.; Caverlee, J.; and Liu, H. 2022a. Meta Propagation Networks for Graph Few-shot Semi-supervised Learning. In *AAAI*.

Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; and Liu, H. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *CIKM*.

Ding, K.; Xu, Z.; Tong, H.; and Liu, H. 2022b. Data augmentation for deep graph learning: A survey. *SIGKDD Explorations*.

Ding, K.; Zhang, C.; Tang, J.; Chawla, N.; and Liu, H. 2022c. Toward Graph Minimally-Supervised Learning. In *KDD*.

Du, Y.; Wang, S.; Guo, X.; Cao, H.; Hu, S.; Jiang, J.; Varala, A.; Angirekula, A.; and Zhao, L. 2021. GraphGT: Machine Learning Datasets for Deep Graph Generation and Transformation. In *NeurIPS*.

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph random neural networks for semi-supervised learning on graphs. In *NeurIPS*.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *NeurIPS*.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*.

Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; and Pan, S. 2021. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In *IJCAI*.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Kulis, B.; and Jordan, M. I. 2011. Revisiting k-means: New algorithms via Bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. 2021. Prototypical contrastive learning of unsupervised representations. In *ICLR*.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*.

Mo, Y.; Peng, L.; Xu, J.; Shi, X.; and Zhu, X. 2022. Simple Unsupervised Graph Representation Learning. AAAI.

Namata, G.; London, B.; Getoor, L.; Huang, B.; and EDU, U. 2012. Query-driven active surveying for collective classification. In *Workshop on MLG*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; and Zhang, C. 2018. Adversarially Regularized Graph Autoencoder for Graph Embedding. In *IJCAI*.

Park, J.; Lee, M.; Chang, H. J.; Lee, K.; and Choi, J. Y. 2019. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *ICCV*, 6519–6528.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *WWW*.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD*.

Resnik, P.; and Hardisty, E. 2010. Gibbs sampling for the uninitiated. *Technical report, Maryland Univ College Park Inst for Advanced Computer Studies*.

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

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of Graph Neural Network Evaluation. In *NeurIPS Relational Representation Learning Workshop*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *ICLR*.

Wang, J.; Ding, K.; Hong, L.; Liu, H.; and Caverlee, J. 2020. Next-item recommendation with sequential hypergraphs. In *SIGIR*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *ICML*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. In *NeurIPS*.

Zhang, C.; Ding, K.; Li, J.; Zhang, X.; Ye, Y.; Chawla, N. V.; and Liu, H. 2022. Few-Shot Learning on Graphs: A Survey. In *IJCAI*.

Zheng, S.; Zhu, Z.; Zhang, X.; Liu, Z.; Cheng, J.; and Zhao, Y. 2020. Distribution-induced bidirectional generative adversarial network for graph representation learning. In *CVPR*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020a. Deep Graph Contrastive Representation Learning. In *ICML Workshop*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020b. Graph Contrastive Learning with Adaptive Augmentation. In *TheWebConf*.