BAYESIAN ROBUST GRAPH CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) have been widely used to learn node representations and with outstanding performance on various tasks such as node classification. However, noise, which inevitably exists in real-world graph data, would considerably degrade the performance of GNNs revealed by recent studies. In this work, we propose a novel and robust method, Bayesian Robust Graph Contrastive Learning (BRGCL), which trains a GNN encoder to learn robust node representations. The BRGCL encoder is a completely unsupervised encoder. Two steps are iteratively executed at each epoch of training the BRGCL encoder: (1) estimating confident nodes and computing robust cluster prototypes of node representations through a novel Bayesian nonparametric method; (2) prototypical contrastive learning between the node representations and the robust cluster prototypes. Experiments on public benchmarks demonstrate the superior performance of BRGCL and the robustness of the learned node representations. The code of BRGCL is available at https://anonymous.4open.science/r/BRGCL-code-2FD9/.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become popular tools for node representation learning in recent years (Kipf & Welling, 2017; Bruna et al., 2014; Hamilton et al., 2017; Xu et al., 2019). Most prevailing GNNs (Kipf & Welling, 2017; Zhu & Koniusz, 2020) leverage the graph structure and obtain the representation of nodes in a graph by utilizing the features of their connected nodes. Benefiting from such propagation mechanism, node representations obtained by GNN encoders have demonstrated superior performance on various downstream tasks such as semi-supervised node classification and node clustering.

Although GNNs have achieved great success in node representation learning, current GNN approaches do not consider the noise in the input graph. However, noise inherently exists in the graph data for many real-world applications. Such noise may be present in node attributes or node labels, which forms two types of noise, attribute noise and label noise. Recent works, such as (Patrini et al., 2017), have evidenced that noisy inputs hurt the generalization capability of neural networks. Moreover, noise in a subset of the graph data can easily propagate through the graph topology to corrupt the remaining nodes in the graph data. Nodes that are corrupted by noise or falsely labeled would adversely affect the representation learning of themselves and their neighbors.

While manual data cleaning and labeling could be remedies to the consequence of noise, they are expensive processes and difficult to scale, thus not able to handle an2 almost infinite amount of noisy data online. Therefore, it is crucial to design a robust GNN encoder that could make use of noisy training data while circumventing the adverse effect of noise. In this paper, we propose a novel and robust method termed Bayesian Robust Graph Contrastive Learning (BRGCL) to improve the robustness of node representations for GNNs. Our key observation is that there exist a subset of nodes which are confident in their class/cluster labels. Usually, such confident nodes are far away from the class/cluster boundaries, so these confident nodes are trustworthy, and noise in these nodes would not degrade the value of these nodes in training a GNN encoder. To infer such confident nodes, we propose a novel algorithm named Bayesian nonparametric Estimation of Confidence (BEC). Since the BRGCL encoder is completely unsupervised, it first infers pseudo labels of all the nodes with a Bayesian nonparametric method only based on the input node attributes, without knowing the ground truth labels or the ground truth class number in the training data. Then, BEC is used to estimate the confident nodes based on the pseudo labels and the graph structure. The robust prototype representations, as the cluster centers of the confident nodes, are computed and used to

train the BRGCL encoder with a loss function for prototypical contrastive learning. The confident nodes are updated during each epoch of the training of the BRGCL encoder, so the robust prototype representations are also updated accordingly.

1.1 CONTRIBUTIONS

Our contributions are as follows.

First, we propose Bayesian Robust Graph Contrastive Learning (BRGCL), where a *fully unsupervised* encoder is trained on noisy graph data. The fully unsupervised BRGCL encoder is trained only on the input node attributes without ground truth labels or even the ground truth class number in the training data. GRGCL leverages confident nodes, which are estimated by a novel algorithm termed Bayesian nonparametric Estimation of Confidence (BEC), to harvest noisy graph data without being compromised by the noise. Experimental results on popular graph datasets evidence the advantage of BRGCL over competing GNN methods for node classification and node clustering on noisy graph data. The significance of the improvement of BRGCL is evidenced by p-values of t-test.

Second, our study reveals the importance of confident nodes in training GNN encoders on noisy graph data, which opens the door for future research in this direction. The visualization results in Section 5.3 show that the confident nodes estimated by BEC are usually far away from the class/cluster boundaries, and so are the robust prototype representations. As a result, the BRGCL encoder trained with such robust prototypes is not vulnerable to noise, and it even outperforms GNNs trained with ground truth labels. The better spectrum of the Neural Tangent Kernel (Jacot et al., 2018) of BRGCL is also demonstrated against its baseline in Section 5.3, explaining BRGCL's better generalization capability from a perspective of spectral analysis of neural networks.

2 RELATED WORKS

Graph Neural Networks. Graph neural networks (GNNs) have become popular tools for node representation learning. They have shown superior performance in various graph learning tasks, such as node classification, node clustering, and graph classification. Given the difference in the convolution domain, current GNNs fall into two classes. The first class features spectral convolution (Bruna et al., 2014; Kipf & Welling, 2017), and the second class (Hamilton et al., 2017; Veličković et al., 2017; Xu et al., 2019) generates node representations by sampling and propagating features from their neighborhood. GNNs such as ChebNet (Bruna et al., 2014) perform convolution on the graph Fourier transforms termed spectral convolution. Graph Convolutional Network (GCN) (Kipf & Welling, 2017) further simplifies the spectral convolution (Bruna et al., 2014) by its first-order approximation. GNNs such as GraphSAGE (Hamilton et al., 2017) propose to learn a function that generates node representations by sampling and propagating features from a node's connected neighborhood to itself. Various designs of the propagation function have been proposed. For instance, Graph Attention Network (GAT) (Veličković et al., 2017) proposes to learn masked self-attention layers that enable nodes to attend over their neighborhoods' features. Different from GNNs based on spectral convolution, such methods could be trained on mini-batches (Hamilton et al., 2017; Xu et al., 2019), so they are more scalable to large graphs.

However, as pointed out by (Dai et al., 2021), the performance of GNNs can be easily degraded by noisy training data (NT et al., 2019). Moreover, the adverse effects of noise in a subset of nodes can be exaggerated by being propagated to the remaining nodes through the network structure, exacerbating the negative impact of noise.

Existing Methods Handing Noisy Data. Previous works (Zhang et al., 2021) have shown that deep neural networks usually generalize badly when trained on input with noise. Existing literature on robust learning with noisy inputs mostly focuses on image or text domain. Such robust learning methods fall into two categories. The first category (Patrini et al., 2017; Goldberger & Ben-Reuven, 2016) mitigates the effects of noisy inputs by correcting the computation of loss function, known as loss corruption. The second category aims to select clean samples from noisy inputs for the training (Malach & Shalev-Shwartz, 2017; Jiang et al., 2018; Yu et al., 2019; Li et al., 2020; Han et al., 2018), known as sample selection. For example, (Goldberger & Ben-Reuven, 2016) corrects the predicted probabilities with a corruption matrix computed on a clean set of inputs. On the other hand, recent sample selection methods usually select a subset of training data to perform robust learning.

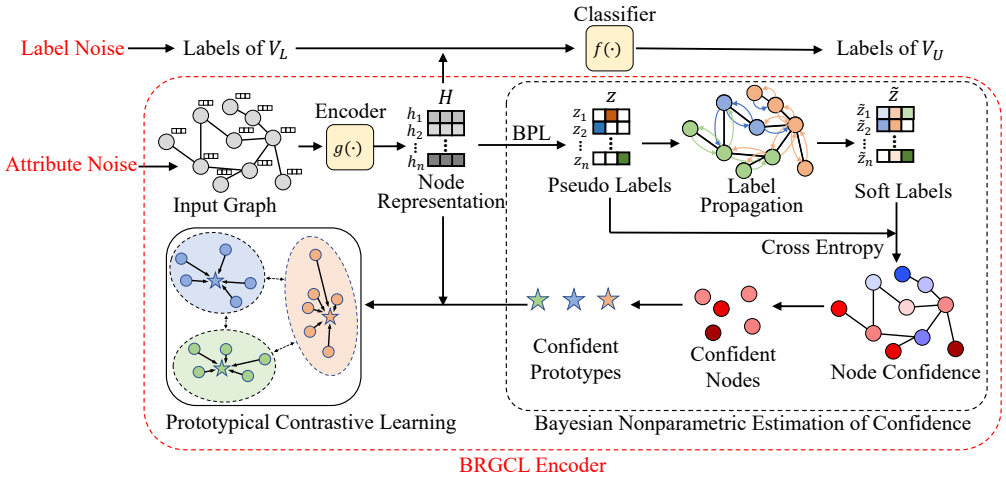


Figure 1: Illustration of the BRGCL encoder. BPL stands for the Bayesian nonparametric Prototype Learning to be introduced in Section 4.2, a Bayesian nonparametric algorithm to estimate the pseudo labels of nodes. In the illustration of confident nodes, more confident nodes are marked in more red, and less confident nodes are marked in more blue.

Among the existing loss correction and sample selection methods, Co-teaching (Han et al., 2018) is promising, which trains two deep neural networks and performs sample selection in a training batch by comparing predictions from the two networks. However, such sample selection strategy does not generalize well in graph domain (Dai et al., 2021) due to the extraordinarily small size of labeled nodes. More details are to be introduced in Section 4.2. Self-Training (Li et al., 2018) finds nodes with the most confident pseudo labels, and it augments the labeled training data by incorporating confident nodes with their pseudo labels into the existing training data. In addition to the above two categories of robust learning methods, recent studies (Kang et al., 2020; Zhong et al., 2021; Wang et al., 2021) show that decoupling the feature representation learning and the training of the classifier can also improve the robustness of the learned feature representation.

3 Problem Setup

An attributed graph consisting of N nodes is 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. $\mathbf{X} \in \mathbb{R}^{N \times D}$ are the node attributes, and the attributes of each node is in $\mathbb{R}^d$. Let $\mathbf{A} \in \{0, 1\}^{N \times N}$ be the adjacency matrix of graph $\mathcal{G}$, with $\mathbf{A}_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ denotes the adjacency matrix for a graph with self-loops added. $\tilde{\mathbf{D}}$ denotes the diagonal degree matrix of $\tilde{\mathbf{A}}$. Let $\mathcal{V}_\mathcal{L}$ and $\mathcal{V}_\mathcal{U}$ denote the set of labeled nodes and unlabeled nodes, respectively.

Noise usually exists in the input node attributes or labels of real-world graphs, which degrades the quality of the node representation obtained by common GCL encoders and affects the performance of the classifier trained on such representations. We aim to obtain node representations robust to noise in two cases, where noise is present in either the labels of $\mathcal{V}_\mathcal{L}$ or in the input node attributes $\mathbf{X}$. That is, we consider either noisy label or noisy input node attributes.

The goal of BRGCL is to learn a node representation $\mathbf{H} = g(\mathbf{X}, \mathbf{A})$, such that the node representations $\{\mathbf{h}_i\}_{i=1}^N$ are robust to noise in the above two cases, where $g(\cdot)$ is the BRGCL encoder. In our work, we use a two-layer GCN as our encoder. Thus $g(\mathbf{X}, \mathbf{A}) = \sigma(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)})$, where $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$. $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are the weight matrices. σ is the activation function ReLu. $\mathbf{h}_i$ is the i-th row of $\mathbf{H}$. To evaluate the performance of the robust node representations by BRGCL, the node representations $\{\mathbf{h}_i\}_{i=1}^N$ are used for the following two tasks.

(1) Semi-supervised node classification, where a classifier is trained on $\mathcal{V}_\mathcal{L}$, and then the classifier predicts the labels of the remaining unlabeled nodes, $\mathcal{V}_\mathcal{U}$.

(2) Node clustering, where K-means clustering is performed on the node representations $\{\mathbf{h}_i\}_{i=1}^N$ to obtain node clusters.

Notations. Throughout this paper, we use $\|\cdot\|_2$ to denote the Euclidean norm of a vector and $[n]$ to denote all the natural numbers between 1 and n inclusively.

4 BAYESIAN ROBUST GRAPH CONTRASTIVE LEARNING

We propose Bayesian Robust Graph Contrastive Learning (BRGCL) in this section to improve the robustness of node representations. First, we review the preliminaries of graph contrastive learning. Next, we propose Bayesian nonparametric Estimation of Confidence (BEC) algorithm to estimate robust nodes and prototypes. Then, we show details of node classification and node clustering. At last, we propose a decoupled training pipeline of BRGCL for semi-supervised node classification. Figure 1 illustrates the overall framework of our proposed BRGCL.

4.1 PRELIMINARY OF GRAPH CONTRASTIVE LEARNING

The general node representation learning aims to train an encoder $g(\cdot)$, which is a two-layer Graph Convolution Neural Network (GCN) (Kipf & Welling, 2017), to generate discriminative node representations. In our work, we adopt contrastive learning to train the BRGCL encoder $g(\cdot)$. To perform contrastive learning, two different views, denoted as $G^1 = (\mathbf{X}^1, \mathbf{A}^1)$ and $G^2 = (\mathbf{X}^2, \mathbf{A}^2)$ are generated by node dropping, edge perturbation, and attribute masking. The representation of two generated views are denoted as $\mathbf{H}^1 = g(\mathbf{X}^1, \mathbf{A}^1)$ and $\mathbf{H}^2 = g(\mathbf{X}^2, \mathbf{A}^2)$, with $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$ being the i-th row of $\mathbf{H}^1$ and $\mathbf{H}^2$, respectively. It is preferred that the mutual information between $\mathbf{H}^1$ and $\mathbf{H}^2$ is maximized. For computational reason, its lower bound is usually used as the objective for contrastive learning. We use InfoNCE (Li et al., 2021a) as our node-wise contrastive loss, that is, $\mathcal{L}_{node} = \sum_{i=1}^N -\log \frac{s(\mathbf{h}_i^1, \mathbf{h}_i^2)}{s(\mathbf{h}_i^1, \mathbf{h}_i^2) + \sum_{j=1}^N s(\mathbf{h}_i^1, \mathbf{h}_j^2)}$, where $s(\mathbf{h}_i^1, \mathbf{h}_i^2) = \frac{|\langle \mathbf{h}_i^1, \mathbf{h}_i^2 \rangle|}{\|\mathbf{h}_i^1\|_2 \|\mathbf{h}_i^2\|_2}$ is the cosine similarity between two node representations, $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$.

In addition to the node-wise contrastive learning, we also adopt prototypical contrastive learning (Li et al., 2021a) to capture semantic information in the node representations, which can be interpreted as maximizing the mutual information between node representation and a set of estimated cluster prototypes $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_K\}$. Here K is the number of cluster prototypes. The loss function for prototypical contrastive learning is $\mathcal{L}_{proto} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\mathbf{h}_i \cdot \mathbf{c}_k / \tau)}{\sum_{k=1}^K \exp(\mathbf{h}_i \cdot \mathbf{c}_k / \tau)}$.

BRGCL aims to improve the robustness of node representations by prototypical contrastive learning. Our key observation is that there exists a subset of nodes that are confident about their class/cluster labels because they are faraway from class/cluster boundaries. We propose an effective method to infer such confident nodes. Because the BRGCL encoder is completely unsupervised, it does not have access to the ground truth label or ground truth class/cluster number. Therefore, our algorithm for selection of confident nodes is based on Bayesian non-parameter style inference, and the algorithm is termed Bayesian nonparametric Estimation of Confidence (BEC) to be introduced next.

4.2 BAYESIAN NONPARAMETRIC ESTIMATION OF CONFIDENCE (BEC)

The key idea of Bayesian nonparametric Estimation of Confidence (BEC) is to estimate robust nodes by the confidence of nodes in their labels. Intuitively, nodes more confident in their labels are less likely to be adversely affected by noise. Because BRGCL is unsupervised, pseudo labels are used as the labels for such estimation.

We propose Bayesian nonparametric Prototype Learning (BPL) to infer the pseudo labels of nodes. BPL, as a Bayesian nonparametric algorithm, infers the cluster prototypes by the Dirichlet Process Mixture Model (DPMM) under the assumption that the distribution of node representations is a mixture of Gaussians. The Gaussians share the same fixed covariance matrix $\sigma\mathbf{I}$, and each Gaussian is used to model a cluster. The DPMM model is specified by

$$G \sim \text{DP}(G_0, \alpha), \boldsymbol{\phi}_i \sim G, \mathbf{h}_i \sim \mathcal{N}(\boldsymbol{\phi}_i, \sigma\mathbf{I}), \quad i = 1, ..., N, \tag{1}$$

where G is a Gaussian distribution draw from the Dirichlet process $\text{DP}(G_0, \alpha)$, and α is the concentration parameter for $\text{DP}(G_0, \alpha)$. ϕ_i is the mean of the Gaussian sampled for generating the node

representation $\mathbf{h}_i$. G_0 is the prior over means of the Gaussians. G_0 is set to a zero-mean Gaussian $\mathcal{N}(\mathbf{0}, \rho\mathbf{I})$ for $\rho > 0$. A collapsed Gibbs sampler (Resnik & Hardisty, 2010) is used to infer the components of the GMM with the DPMM. 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 & Jordan, 2011), such a process is almost equivalent to K-means when σ, the variance of the Gaussians, goes to 0. The almost zero variance eliminates the need to estimate the variance σ, making the inference efficient.

Let $\tilde{K}$ denote the number of inferred prototypes at the current iteration, the pseudo label z_i of node v_i is then calculated by

$$z_i = \arg\min_k \{d_{ik}\}, \ i = 1, ..., N, \quad d_{ik} = \begin{cases} \|\mathbf{h}_i - \mathbf{c}_k\|_2^2 & k = 1, ..., \tilde{K}, \\ \xi & k = \tilde{K} + 1, \end{cases} \tag{2}$$

where the Euclidean distance $\{d_{ik}\}$ is used to determine the pseudo labels of the node representation $\mathbf{h}_i$. ξ is the margin to initialize a new prototype. In practice, we choose the value of ξ by performing cross-validation on each dataset with details in Section A in the supplementary.

After obtaining the pseudo labels of nodes by BPL with K being the inferred number of prototypes, we estimate the confidence of the nodes based on their pseudo labels and the graph structure. We first select nodes confident in their labels, also referred to as confident nodes, by considering the label information from the neighborhood of each node specified by the adjacency matrix. Let $\boldsymbol{z}_i$ denote the one-hot pseudo label of node v_i estimated by the Bayesian method. Label propagation (Zhang & Chen, 2018) is applied based on the adjacency matrix to get a soft pseudo label for each node. Let $\mathbf{Z} \in \mathbb{R}^{N \times K}$ be the matrix of pseudo labels with $\mathbf{z}_i$ being the i-th row of $\mathbf{Z}$. The label propagation runs the following update for T steps,

$$\mathbf{Z}^{(t+1)} = (1 - \alpha)\tilde{\mathbf{A}}\mathbf{Z}^{(t)} + \alpha\mathbf{Z} \quad t = 1, ..., T - 1, \tag{3}$$

where T is the number of propagation steps, α is the teleport probability, which are set to the suggested values in (Zhang & Chen, 2018). Let $\tilde{\mathbf{Z}} = \mathbf{Z}^{(T)}$ be the soft labels obtained by the label propagation with $\tilde{\boldsymbol{z}}_i$ being the i-th row of $\tilde{\mathbf{Z}}$. Following (Han et al., 2018), we use the cross-entropy between $\boldsymbol{z}_i$ and $\tilde{\boldsymbol{z}}_i$, denoted by $\phi(\boldsymbol{z}_i, \tilde{\boldsymbol{z}}_i)$, to identify confident nodes. Intuitively, smaller cross-entropy $\phi(\boldsymbol{z}_i, \tilde{\boldsymbol{z}}_i)$ of a node v_i leads to a larger probability of the pseudo label, so node v_i is more confident about its pseudo label $\tilde{\boldsymbol{z}}_i$. As a result, we denote the set of confident nodes assigned to the k-th cluster as

$$\mathcal{T}_k = \{\boldsymbol{h}_i \mid \phi(\boldsymbol{z}_i, \tilde{\boldsymbol{z}}_i) < \gamma_k\}, \tag{4}$$

where γ_k is a threshold for the k-th class. Figure 3 illustrates the cross-entropy values of all the nodes for the case that different levels of noise are present in the input node attributes, where the heat value indicates the corresponding cross-entropy value for every node. The confident nodes with less cross-entropy values, which are marked in more red, are far away from cluster boundaries, so that noise on these nodes is more unlikely to affect their classification/clustering labels. These confident nodes are the robust nodes leveraged by BRGCL to fight against noise.

The threshold γ_k is dynamically set by

$$\gamma_k = 1 - \min\left\{\gamma_0, \gamma_0 \frac{t}{t_{\mathrm{m}}}\right\}, \tag{5}$$

where t is the current epoch number and t_{m} is a preset number of epochs. In Co-teaching (Han et al., 2018), a similar threshold is used to select a ratio of data for training. However, due to the limited size of training data in graph domain, training with only a subset of nodes usually leads to degraded performance. For example, with 5% of nodes labeled on Cora dataset, only 1% of nodes will be used for training if the threshold is set to 20% by Co-teaching. In contrast, BEC selects confident nodes by a dynamic threshold on the confidence of nodes in their labels given the labels from their neighbors. The selected confident nodes are only used to obtain the robust prototype representations, and BRGCL is trained with such robust prototypes to obtain robust representations for all the nodes of the graph.

γ is an annealing factor. In practice, the value of γ_0 is decided by cross-validation for each dataset, with details in Section A of the supplementary. Previous methods such as (Li et al., 2021a) estimate each prototype as the mean of node representations assigned to that prototype. After acquiring the confident nodes $\{\mathcal{T}_k\}_{k=1}^K$, the prototype representations are updated by $\boldsymbol{c}_k = \frac{1}{|\mathcal{T}_k|} \sum_{\boldsymbol{h}_i \in \mathcal{T}_k} \boldsymbol{h}_i$ for

Algorithm 1 Training algorithm of BRGCL encoder

Input: The input attribute matrix $\mathbf{X}$, adjacency matrix $\mathbf{A}$, and the training epochs $t_{\max}$.
Output: The parameter of BRGCL encoder g.

```
Initialize the parameter of BRGCL encoder g
for t ← 1 to t_max do
    Calculate node representations by H = g(X, A)
    Generate augmented views G^1 = (X^1, A^1) and G^2 = (X^2, A^2)
    Calculate node representations of augmented views by H^1 = g(X^1, A^1) and H^2 = g(X^2, A^2)
    Calculate loss L_node
    Obtain the pseudo labels of all the nodes Z and the number of inferred prototypes K by Eq. (2)
    Obtain soft labels of nodes Z̃ by label propagation in Eq. (3)
    Update the confidence thresholds {γ_k}_{k=1}^K by Eq. (5)
    Estimate the sets of confident nodes {T_k}_{k=1}^K by Eq. (4)
    Update confident prototype representations by c_k = 1/|T_k| Σ_{h_i ∈ T_k} h_i for all k ∈ [K]
    Update the parameter of BRGCL encoder g using the loss L_rep in Eq. (6)
end for
return The BRGCL encoder g
```

each $k \in [K]$. With the updated cluster prototypes $\{\boldsymbol{c}_k\}_{k=1}^K$ in the prototypical contrastive learning loss $\mathcal{L}_{proto}$, we train the encoder $g(\cdot)$ with the following overall loss function,

$$\mathcal{L}_{rep} = \mathcal{L}_{node} + \mathcal{L}_{proto}. \tag{6}$$

Training BRGCL with the loss function $\mathcal{L}_{rep}$ does not require any information about the ground truth labels. We summarize the training algorithm for the BRGCL encoder in Algorithm 1. It is noted that confident nodes and robust prototypes are estimated at each epoch by BEC.

4.3 Decoupled Training

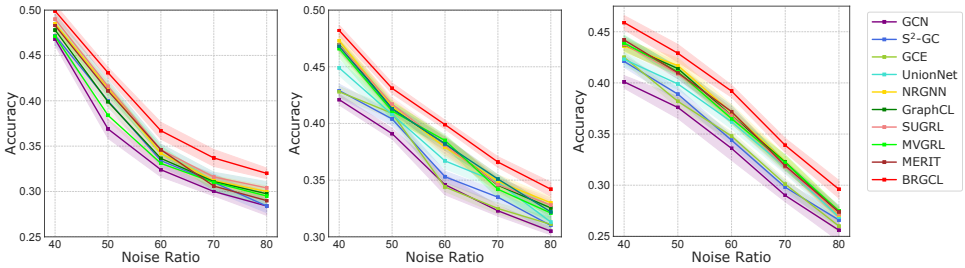
The typical pipeline for semi-supervised node classification is to jointly train the classifier and the encoder. However, the noise in the training data would degrade the performance of the classifier. To alleviate this issue, we decouple the representation learning for the nodes from the classification of nodes to mitigate the effect of noise, which consists of two steps. In the first step, the BRGCL encoder $g(\cdot)$ is trained by Algorithm 1. In the second step, with the node representation $\mathbf{H}$ from the trained BRGCL encoder, the classifier $f(\cdot)$ is trained by optimizing the loss function $\mathcal{L}_{\text{cls}}$. In Section B.5 of the supplementary, we show the advantage of such decoupled learning pipeline over the conventional joint training of encoder and classifier.

5 Experiments

In this section, we evaluate the performance of BRGCL on five public benchmarks, with details deferred to Section A.1 of the supplementary. For semi-supervised node classification, the performance of BRGCL is evaluated with noisy label or noisy input node attributes. For node clustering, only noisy input node attributes are considered because there are no ground truth labels given for clustering purposes. The implementation details about node classification are deferred to Section A.2 of the supplementary.

5.1 Experimental Settings

Due to the fact that most public benchmark graph datasets do not come with corrupted labels or attribute noise, we manually inject noise into public datasets to evaluate our algorithm. We follow the commonly used label noise generation methods from the existing work (Han et al., 2020) to inject label noise. We generate noisy labels over all classes according to a noise transition matrix $Q^{K\times K}$, where Q_{ij} is the probability of nodes from class i being flipped to class j. We consider two types of noise: (1) **Symmetric**, where nodes from each class can be flipped to other classes with a uniform random probability, s.t. $Q_{ij} = Q_{ji}$; (2) **Asymmetric**, where mislabeling only occurs between similar classes. The percentage of nodes with flipped labels is defined as the label noise level in our experiments. To evaluate the performance of our method with attribute noise, we randomly



(a) attribute noise (b) symmetric label noise (c) asymmetric label noise

Figure 2: Performance comparisons on semi-supervised node classification on ogbn-arxiv with different levels of attribute noise, symmetric label noise, and asymmetric label noise. The shaded areas around the lines denote the standard deviation of the classification accuracy.

shuffle a certain percentage of input attributes for each node following (Ding et al., 2022). The percentage of shuffled attributes is defined as the attribute noise level in our experiments.

We compare BRGCL against semi-supervised node representation learning methods GCN (Kipf & Welling, 2017), GCE (Zhang & Sabuncu, 2018), S^2GC (Zhu & Koniusz, 2020), UnionNet (Li et al., 2021b), NRGNN (Dai et al., 2021). In addition, we also compare BRGCL against state-of-the-art GCL methods, including GraphCL (You et al., 2020), MVGRL (Hassani & Khasahmadi, 2020), MERIT (Jin et al., 2021), and SUGRL (Mo et al., 2022). The training settings for different baselines are categorized into two setups: (1) **Unsupervised Setup**, where the training of the encoder does not use the ground truth label information. The node representations obtained by the encoder are then used for downstream tasks, which are node classification and node clustering; (2) **Supervised Setup**, where the training of the encoder uses the ground truth label information. Our proposed BRGCL follows the unsupervised setup in all our experiments, and every baseline follows its corresponding setup by its nature.

5.2 Evaluation Results

Semi-supervised Node Classification with Label Noise. We compare BRGCL against competing methods for semi-supervised node classification on input with two types of label noise. To show the robustness of BRGCL against label noise, we perform the experiments on graphs injected with different levels of label noise ranging from 40% to 80% with a step of 20%. The classification follows the widely used semi-supervised setting (Kipf & Welling, 2017). Note the labels are only used for the training of the classifier. The BRGCL encoder generates node representations, and the classifier for node classification is trained on these node representations.

In our experiment, a two-layer MLP whose hidden dimension is 128 is used as the classifier. The results of different methods with respect to different levels of symmetric and asymmetric label noise on ogbn-arxiv are illustrated in Figure 2. Detailed results on PubMed and ogbn-arxiv are shown in Table 1, where we report the means of the accuracy of 10 runs and the standard deviation. We also show the results on Cora, Citeseer, and Coauthor CS in Section B.1 in the supplementary. It is observed from the results that BRGCL outperforms all the baselines, including the methods using ground truth labels to train their encoders. By selecting confident nodes and computing robust prototypes using BEC, BRGCL outperforms all the baselines by an even larger margin with a larger label noise level. To verify the statistical significance of improvements, we show the p-values of t-test between BRGCL and the second best baseline in Section B.2 in the supplementary. The p-values for all datasets with all noise levels for both symmetric label noise and asymmetric label noise are less than 0.05, suggesting the statistically significant improvement of BRGCL over baseline methods.

Semi-supervised Node Classification with Attribute Noise. We compare BRGCL with baselines for noisy input with attribute noise levels ranging from 40% to 80% with a step of 20%. The results on ogbn-arxiv are illustrated in Figure 2. Detailed results on PubMed, and ogbn-arxiv are shown in Table 1, where we report the means of the accuracy of 10 runs and the standard deviation. The results on Cora, Citeseer, and Coauthor CS are deferred to Section B.1 in the supplementary. The results clearly show that BRGCL is more robust to attribute noise compared to all the baselines for different

Table 1: Performance comparison for node classification on PubMed and ogbn-arxiv with asymmetric label noise, symmetric label noise, and attribute noise. Results with label noise on Cora, Citeseer, and Coauthor CS, and results with attribute noise for all the datasets are deferred to Section B.1 in the supplementary. The encoders of methods marked with * are trained with label information. The p-values of the t-test between BRGCL and the second best baseline are attached in Section B.2 in the supplementary.

Dataset	Methods	Noise Level									
		0	40			60			80		
		-	Asymmetric	Symmetric	Attribute	Asymmetric	Symmetric	Attribute	Asymmetric	Symmetric	Attribute
PubMed	GCN *	0.790±0.007	0.584±0.022	0.574±0.012	0.595±0.012	0.405±0.025	0.386±0.011	0.488±0.013	0.305±0.022	0.295±0.013	0.423±0.013
	S^2GC *	0.799±0.005	0.585±0.023	0.589±0.013	0.610±0.009	0.421±0.030	0.401±0.014	0.497±0.012	0.310±0.039	0.290±0.019	0.431±0.010
	GCE	0.792±0.009	0.589±0.018	0.581±0.011	0.590±0.014	0.430±0.012	0.399±0.012	0.491±0.010	0.311±0.021	0.301±0.011	0.424±0.012
	UnionNET	0.793±0.008	0.603±0.020	0.620±0.012	0.592±0.012	0.445±0.022	0.424±0.013	0.489±0.015	0.313±0.025	0.327±0.015	0.435±0.009
	NRGNN	0.797±0.008	0.602±0.022	0.618±0.013	0.603±0.008	0.443±0.012	0.434±0.012	0.499±0.009	0.330±0.023	0.325±0.013	0.433±0.011
	GraphCL	0.790±0.006	0.592±0.016	0.603±0.015	0.601±0.007	0.434±0.015	0.418±0.019	0.479±0.014	0.310±0.017	0.302±0.014	0.439±0.013
	SUGRL	0.819±0.005	0.603±0.013	0.615±0.013	0.615±0.010	0.445±0.011	0.441±0.011	0.501±0.007	0.321±0.009	0.321±0.009	0.446±0.010
	MVGRL	0.794±0.003	0.599±0.012	0.613±0.012	0.606±0.008	0.441±0.013	0.433±0.013	0.496±0.010	0.322±0.012	0.312±0.012	0.438±0.010
	MERIT	0.801±0.004	0.593±0.011	0.612±0.011	0.613±0.011	0.447±0.012	0.443±0.012	0.497±0.009	0.328±0.011	0.323±0.011	0.445±0.009
	BRGCL	0.793±0.007	**0.624±0.014**	**0.632±0.010**	**0.625±0.011**	**0.465±0.011**	**0.468±0.010**	**0.514±0.011**	**0.342±0.014**	**0.349±0.013**	**0.470±0.011**
ogbn-arxiv	GCN *	0.717±0.003	0.401±0.014	0.421±0.014	0.478±0.010	0.336±0.011	0.346±0.021	0.339±0.012	0.286±0.022	0.256±0.010	0.294±0.013
	S^2GC *	0.732±0.003	0.417±0.017	0.429±0.014	0.492±0.010	0.344±0.016	0.353±0.031	0.343±0.009	0.297±0.023	0.266±0.013	0.284±0.012
	GCE	0.720±0.004	0.410±0.018	0.428±0.008	0.480±0.014	0.348±0.019	0.344±0.019	0.342±0.015	0.310±0.014	0.260±0.011	0.275±0.015
	UnionNET	0.724±0.006	0.429±0.021	0.449±0.007	0.485±0.012	0.362±0.018	0.367±0.008	0.340±0.009	0.332±0.019	0.269±0.013	0.280±0.012
	NRGNN	0.721±0.006	0.449±0.014	0.466±0.009	0.485±0.012	0.371±0.020	0.379±0.008	0.342±0.011	0.330±0.018	0.271±0.018	0.300±0.010
	GraphCL	0.701±0.004	0.431±0.013	0.455±0.009	0.467±0.013	0.364±0.016	0.373±0.012	0.328±0.010	0.317±0.022	0.266±0.015	0.294±0.012
	SUGRL	0.693±0.002	0.439±0.010	0.467±0.010	0.480±0.012	0.365±0.013	0.385±0.011	0.341±0.009	0.327±0.011	0.275±0.011	0.295±0.011
	MVGRL	0.713±0.002	0.443±0.009	0.461±0.009	0.481±0.008	0.372±0.012	0.382±0.012	0.339±0.009	0.329±0.013	0.274±0.013	0.290±0.012
	MERIT	0.717±0.001	0.442±0.009	0.463±0.009	0.483±0.010	0.368±0.011	0.381±0.011	0.341±0.012	0.324±0.012	0.272±0.010	0.304±0.009
	BRGCL	0.720±0.005	**0.459±0.013**	**0.482±0.006**	**0.495±0.010**	**0.392±0.014**	**0.399±0.009**	**0.350±0.011**	**0.345±0.012**	**0.296±0.013**	**0.326±0.012**

noise levels. To verify the statistical significance of improvements, we show the p-values of t-test between BRGCL and the second best baseline in Section B.2 in the supplementary. The p-values for all datasets with all levels of attribute noise are less than 0.05, suggesting the statistically significant improvement of BRGCL over baseline methods.

Node Clustering with Attribute Noise. To further evaluate the robustness of node representation learned by BRGCL, we perform experiments on node clustering with attribute noise injected. We follow the same evaluation protocol as that in (Hassani & Khasahmadi, 2020). K-means is applied on the learned node representations to obtain clustering results. We use accuracy (ACC) and normalized mutual information (NMI) as the performance metrics for clustering. The node clustering results for inputs with 60% attribute noise are shown in Table 2. We report the averaged clustering results and standard deviations over 20 times of execution. Results on clean benchmark datasets are deferred to Section B.3 in the supplementary. It is observed that node representation obtained by BRGCL is more robust to attribute noise for node clustering. To show the statistical significance of improvements, we also calculate p-values of the t-test between BRGCL and the second best baseline for each result. It is observed that BRGCL significantly improves the performance of node clustering as the p-values for ACC and NMI are less than 0.05 on all datasets.

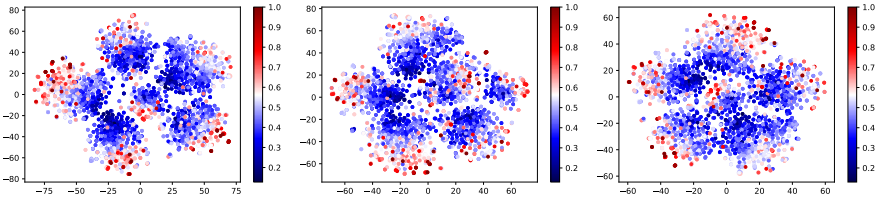
Table 2: Node clustering performance comparison on benchmark datasets with 60% input attribute noise. Results on clean benchmark datasets are shown to Section B.3 in the supplementary. The p-values of the t-test between BRGCL and the second best baseline are listed in the last row of the table.

Methods	Cora		Citeseer		PubMed		Coauthor CS		ogbn-arxiv	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Supervised										
GCN	57.4±0.61	44.7±0.57	57.1±0.65	35.4±0.34	56.9±0.99	28.3±0.46	52.8±0.74	52.6±0.95	43.7±1.19	49.6±0.84
S^2GC	58.4±0.72	47.3±0.79	58.3±0.82	35.1±0.65	57.4±0.89	28.3±0.31	53.6±0.99	54.0±1.09	45.2±0.97	50.2±0.70
NRGNN	61.1±0.73	47.8±0.93	57.8±0.77	36.2±0.71	57.1±1.03	29.1±0.59	53.3±0.87	54.1±1.02	44.1±1.04	50.1±0.85
Unsupervised										
K-means	39.9±0.94	26.9±0.88	44.8±0.59	26.8±1.76	49.0±1.45	29.3±1.49	25.4±1.76	14.6±1.86	24.3±1.76	27.9±1.86
GAE	49.1±0.95	36.9±0.67	33.2±0.64	16.4±1.36	56.6±0.87	26.1±0.65	39.6±1.25	38.9±1.40	34.5±1.14	36.4±1.32
ARVGA	53.8±1.01	39.0±0.59	45.2±0.82	24.2±0.78	57.2±0.69	27.0±0.46	49.8±0.65	48.3±1.13	40.2±0.77	44.3±1.03
GALA	63.3±0.78	50.0±0.68	59.4±0.80	35.8±0.88	57.1±0.79	29.1±0.17	52.5±1.03	53.8±0.98	45.2±0.97	50.5±0.79
GraphCL	61.2±0.96	49.1±0.79	58.3±0.88	34.9±1.02	57.3±0.89	29.1±0.49	53.2±0.88	54.2±1.14	43.9±0.97	49.3±1.03
MVGRL	62.5±0.79	50.5±0.63	59.2±0.79	35.7±0.76	57.6±0.70	29.6±0.55	54.1±0.87	55.2±1.02	45.1±0.89	50.2±0.95
MERIT	63.0±0.87	51.1±0.75	59.2±0.69	36.1±0.45	57.9±0.80	30.2±0.42	54.8±0.87	56.4±0.79	45.4±0.78	51.0±0.81
BRGCL	**63.8±0.69**	**51.9±0.81**	**60.3±0.79**	**37.1±0.63**	**58.8±0.59**	**30.9±0.85**	**56.1±0.64**	**58.2±0.96**	**46.5±0.86**	**52.2±0.91**
p-value	0.0014	0.0021	0.0231	0.0030	0.0401	0.0154	0.0075	0.0102	0.0112	0.0144

Comparison to Existing Sample Selection. We also compare our BRGCL to the representative sample selection methods for node classification, including Co-teaching (Han et al., 2018), in Section B.4 of the supplementary. It is observed that BRGCL outperforms these competing methods by a noticeable margin.

5.3 VISUALIZATION OF CONFIDENCE SCORE AND SPECTRUM OF BRGCL

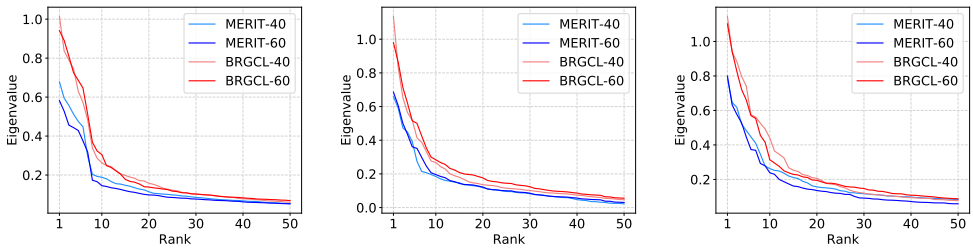
We visualize the confident nodes selected by BEC in the embedding space of the learned node representations in Figure 3. The node representations are visualized by the t-SNE figure. Each mark in t-SNE represents the representation of a node, and the color of the mark denotes the confidence of that node. The results are shown for different levels of attribute noise. It can be observed from Figure 3 that confident nodes, which are redder in Figure 3, are well separated in the embedding space. With a higher level of attribute noise, the bluer nodes from different clusters blended around the cluster boundaries. In contrast, the redder nodes are still well separated and far away from cluster boundaries, which leads to more robustness and better performance in downstream tasks.



(a) noise level = 0 (b) noise level = 20 (c) noise level = 40

Figure 3: Visualization of confident nodes with different levels of attribute noise for semi-supervised node classification on Citeseer.

In order to understand why BRGCL generalizes better than prior SOTA on noisy data, we plot the eigenvalues of the sample Neural Tangent Kernel (Jacot et al., 2018) in Figure 4. Previous studies such as (Rahaman et al., 2019) indicate that neural networks empirically generalize well if the target function lies in low-frequency directions, or the subspace spanned by eigenfunctions corresponding to high eigenvalues of NTK. It can be observed that BRGCL usually has larger eigenvalues than the top baseline, MERIT (Jin et al., 2021), which explains its better generalization from the perspective of spectral analysis of neural networks.



(a) PubMed (b) Coauthor-CS (c) ogbn-arxiv

Figure 4: Illustration of the top-50 eigenvalues of the sample Neural Tangent Kernel for MERIT and BRGCL. Both MERIT and BRGCL are trained on inputs with attribute noise at level 40 and 60.

6 CONCLUSIONS

In this paper, we propose a novel node representation learning method termed Bayesian Robust Graph Contrastive Learning (BRGCL) that aims to improve the robustness of node representations by a novel Bayesian nonparametric algorithm, Bayesian nonparametric Estimation of Confidence (BEC). We evaluate the performance of BRGCL with comparison to competing baselines on semi-supervised node classification and node clustering, where graph data are corrupted with noise in either the labels for the node attributes. Experimental results demonstrate that BRGCL generates more robust node representations with better performance than the current state-of-the-art node representation learning methods.

REFERENCES

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *ICLR*, 2014.

Enyan Dai, Charu Aggarwal, and Suhang Wang. Nrgnn: Learning a label noise-resistant graph neural network on sparsely and noisily labeled graphs. *SIGKDD*, 2021.

Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data augmentation for deep graph learning: A survey. *arXiv preprint arXiv:2202.08235*, 2022.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. pp. 8536–8546, 2018.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, 2020.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *The 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*, 2011.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021a.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

Yayong Li, Jie Yin, and Ling Chen. Unified robust training for graph neural networks against label noise. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 528–540. Springer, 2021b.

Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NeurIPS*, pp. 960–970, 2017.

Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. AAAI, 2022.

Hoang NT, Choong Jin, and Tsuyoshi Murata. Learning graph neural networks with noisy labels. In *2nd ICLR Learning from Limited Labeled Data (LLD) Workshop*, 2019.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pp. 1944–1952, 2017.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical report, Maryland Univ College Park Inst for Advanced Computer Studies, 2010.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1239–1248, 2021.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823, 2020.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, pp. 5171–5181, 2018.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16489–16498, 2021.

Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations*, 2020.

A IMPLEMENTATION DETAILS

A.1 DATASETS

We evaluate BRGCL on five public benchmarks that are widely used for node representation learning, namely Cora, Citeseer, PubMed (Sen et al., 2008), Coauthor CS, and ogbn-arxiv (Hu et al., 2020). Cora, Citeseer and PubMed are three most widely used citation networks. Coauthor CS is co-authorship graph. The ogan-arxiv is a directed citation graph. We summarize the statistics of the datasets in Table 3. Among the five benchmarks, ogbn-arxiv is known for its larger scale, and is more challenging to deal with. For all our experiments, we follow the default separation of training, validation, and test sets on each benchmark.

Table 3: The statistics of the datasets.

Dataset	**Nodes**	**Edges**	**Features**	**Classes**
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Coauthor CS	18,333	81,894	6,805	15
ogbn-arxiv	169,343	1,166,243	128	40

A.2 MORE DETAILS ABOUT NODE CLASSIFICATION

The robust node representations are used to perform node classification and node clustering mentioned in Section 3 of the main paper. More details about node classification are introduced in this subsection. As the connected neighbors in a graph usually show similar semantic information, we generate soft labels of nodes via label propagation on the graph to take the advantage of the information from the neighborhood. The classifier for node classification is trained with soft labels instead of hard labels.

First, we define the one-hot hard label matrix $\mathbf{Y} \in \mathbb{R}^{N\times K}$, where $\mathbf{Y}_{ij} = 1$ if and only if node v_i is in class j for $i \in [N]$ and $j \in [K]$. If a node $v_i \in \mathcal{V}_{\mathcal{L}}$, then $\mathbf{Y}_{ij} = 1$ if the ground truth label of v_i is j. If $v_i \notin \mathcal{V}_{\mathcal{L}}$, we initialize $\mathbf{Y}_{ij} = 0$ for all $j \in [K]$. Then the soft labels of all the nodes are generated by graph label propagation. Similar to (3), after T aggregation steps of label propagation, we have $\mathbf{Y}^{(t+1)} = (1-\beta)\tilde{\mathbf{A}}\mathbf{Y}^{(t)} + \beta\mathbf{Y}^{(0)}, t = 1, ..., T-1$, where $\mathbf{Y}^{(0)} = \mathbf{Y}$, β is the teleport probability. The soft labels are then obtained by $\tilde{\mathbf{Y}} = \text{softmax}(\mathbf{Y}^{(T)})$. We denote the i-th row of $\tilde{\mathbf{Y}}$ by $\tilde{\boldsymbol{y}}_i$, which is the soft label of node v_i. $f(\cdot)$ is a classifier built by a two-layer MLP followed by a softmax function, which is trained by minimizing the standard loss function for classification, $\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{V}_{\mathcal{L}}|}\sum_{v_i\in\mathcal{V}_{\mathcal{L}}} H(\tilde{\boldsymbol{y}}_i, f(\mathbf{h}_i))$, where H is the cross-entropy function.

A.3 TUNING HYPER-PARAMETERS BY CROSS-VALIDATION

In this section, we show the tuning procedures on the hyper-parameters ξ from Equation (2) and γ_0 from Equation (5). We perform cross-validations on 20% of training data to decide the value of ξ and γ_0. The value of ξ is selected from $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0, 5\}$. The value of γ_0 is selected from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. The selected values for ξ and γ_0 on each dataset are shown in Table 4.

Table 4: Selected hyper-parameters for each dataset.

Dataset	Cora	Citeseer	PubMed	Coauthor CS	ogbn-arxiv
ξ	0.20	0.15	0.35	0.40	0.25
γ_0	0.3	0.5	0.7	0.4	0.4

B ADDITIONAL EXPERIMENT RESULTS

B.1 ADDITIONAL NODE CLASSIFICATION RESULTS WITH LABEL NOISE AND ATTRIBUTE NOISE.

The results for node classification with symmetric label noise and asymmetric label noise on Cora, Citeseer, and Coauthor CS are shown in Table 5. In addition, the results for node classification with attribute noise on Cora, Citeseer, and Coauthor CS are shown in Table 6. It is observed that BRGCL outperforms all the baselines for node classification with both label noise and attribute noise on all three benchmark datasets.

Table 5: Performance comparison on node classification with symmetric label noise and asymmetric label noise.

Dataset	Methods	Noise Level			
		0	40	60	80
		-	Asymmetric / Symmetric	Asymmetric / Symmetric	Asymmetric / Symmetric
Cora	GCN *	0.817±0.005	0.547±0.015 / 0.636±0.007	0.405±0.014 / 0.517±0.010	0.265±0.012 / 0.354±0.014
	S^2GC *	0.831±0.002	0.569±0.007 / 0.664±0.007	0.422±0.010 / 0.535±0.010	0.279±0.014 / 0.366±0.014
	GCE	0.819±0.004	0.573±0.011 / 0.652±0.008	0.449±0.011 / 0.509±0.011	0.280±0.013 / 0.353±0.013
	UnionNET	0.820±0.006	0.569±0.014 / 0.664±0.007	0.452±0.010 / 0.541±0.010	0.283±0.014 / 0.370±0.011
	NRGNN	0.822±0.006	0.571±0.019 / 0.676±0.007	0.470±0.014 / 0.548±0.014	0.282±0.022 / 0.373±0.012
	GraphCL	0.815±0.005	0.560±0.011 / 0.661±0.009	0.450±0.017 / 0.541±0.012	0.270±0.018 / 0.368±0.017
	MVGRL	0.829±0.007	0.566±0.009 / 0.672±0.009	0.455±0.014 / 0.545±0.014	0.275±0.014 / 0.379±0.014
	MERIT	0.831±0.006	0.560±0.008 / 0.670±0.008	0.467±0.013 / 0.547±0.013	0.277±0.013 / 0.385±0.013
	SUGRL	**0.834±0.005**	0.564±0.011 / 0.674±0.012	0.468±0.011 / 0.552±0.011	0.280±0.012 / 0.381±0.012
	BRGCL	0.822±0.006	**0.584±0.009 / 0.694±0.007**	**0.484±0.013 / 0.567±0.013**	**0.295±0.012 / 0.394±0.012**
Citeseer	GCN *	0.703±0.005	0.475±0.023 / 0.501±0.013	0.351±0.014 / 0.341±0.014	0.291±0.022 / 0.281±0.019
	S^2GC *	0.727±0.005	0.488±0.013 / 0.528±0.013	0.363±0.012 / 0.367±0.014	0.304±0.024 / 0.284±0.019
	GCE	0.705±0.004	0.490±0.016 / 0.512±0.014	0.362±0.015 / 0.352±0.010	0.309±0.012 / 0.285±0.014
	UnionNET	0.706±0.006	0.499±0.015 / 0.547±0.014	0.379±0.013 / 0.399±0.013	0.322±0.021 / 0.302±0.013
	NRGNN	0.710±0.006	0.498±0.015 / 0.546±0.015	0.382±0.016 / 0.412±0.016	0.336±0.021 / 0.309±0.018
	GraphCL	0.715±0.008	0.479±0.017 / 0.534±0.016	0.373±0.015 / 0.411±0.014	0.331±0.017 / 0.297±0.016
	MVGRL	0.726±0.007	0.491±0.013 / 0.541±0.013	0.379±0.013 / 0.420±0.013	0.341±0.016 / 0.301±0.016
	MERIT	**0.740±0.007**	0.496±0.012 / 0.536±0.012	0.383±0.011 / 0.425±0.011	0.344±0.014 / 0.301±0.014
	SUGRL	0.730±0.005	0.493±0.011 / 0.541±0.011	0.376±0.009 / 0.421±0.009	0.339±0.010 / 0.305±0.010
	BRGCL	0.722±0.004	**0.510±0.013 / 0.569±0.013**	**0.403±0.012 / 0.433±0.014**	**0.359±0.013 / 0.321±0.014**
Coauthor CS	GCN *	0.918±0.001	0.645±0.009 / 0.656±0.006	0.511±0.013 / 0.501±0.009	0.429±0.022 0.389±0.011
	S^2GC*	**0.928±0.001**	0.657±0.012 / 0.663±0.006	0.516±0.013 / 0.514±0.009	0.437±0.020 0.396±0.010
	GCE	0.922±0.003	0.662±0.017 / 0.659±0.007	0.515±0.016 / 0.502±0.007	0.443±0.017 0.389±0.012
	UnionNET	0.918±0.002	0.669±0.023 / 0.671±0.013	0.525±0.011 / 0.529±0.011	0.458±0.015 0.401±0.011
	NRGNN	0.919±0.002	0.678±0.014 / 0.689±0.009	0.545±0.021 / 0.556±0.011	0.461±0.012 0.410±0.012
	GraphCL	0.905±0.005	0.664±0.018 / 0.679±0.014	0.541±0.017 / 0.550±0.015	0.441±0.015 0.396±0.014
	MVGRL	0.913±0.001	0.675±0.008 / 0.685±0.008	0.550±0.014 / 0.560±0.014	0.453±0.013 0.405±0.013
	MERIT	0.924±0.004	0.679±0.011 / 0.689±0.008	0.552±0.014 / 0.562±0.014	0.452±0.013 0.403±0.013
	SUGRL	0.922±0.005	0.675±0.010 / 0.695±0.010	0.550±0.011 / 0.560±0.011	0.449±0.011 0.411±0.011
	BRGCL	0.920±0.003	**0.690±0.012 / 0.710±0.008**	**0.566±0.014 / 0.572±0.011**	**0.461±0.011 / 0.428±0.015**

Table 6: Performance comparison on node classification with attribute noise.

Dataset	Methods	Noise Level					
		0	40	50	60	70	80
Cora	GCN *	0.817±0.005	0.639±0.008	0.510±0.006	0.439±0.012	0.371±0.014	0.317±0.013
	S^2GC *	0.831±0.002	0.661±0.007	0.521±0.008	0.454±0.011	0.371±0.010	0.320±0.013
	NRGNN	0.822±0.006	0.654±0.009	0.517±0.009	0.449±0.014	0.385±0.012	0.322±0.013
	SUGRL	0.829±0.007	0.655±0.011	0.522±0.007	0.445±0.012	0.381±0.011	0.330±0.014
	MVGRL	0.831±0.006	0.671±0.009	0.531±0.008	0.450±0.014	0.385±0.010	0.335±0.009
	MERIT	**0.834±0.005**	0.675±0.009	0.528±0.011	0.452±0.012	0.388±0.012	0.338±0.014
	BRGCL	0.822±0.006	**0.690±0.010**	**0.540±0.010**	**0.469±0.013**	**0.399±0.010**	**0.356±0.011**
Citeseer	GCN *	0.703±0.005	0.529±0.009	0.468±0.012	0.372±0.011	0.313±0.011	0.290±0.014
	S^2GC *	0.727±0.005	0.553±0.008	0.491±0.011	0.390±0.013	0.310±0.012	0.2880±0.011
	NRGNN	0.710±0.006	0.540±0.007	0.501±0.013	0.384±0.014	0.317±0.009	0.287±0.012
	SUGRL	0.726±0.007	0.540±0.008	0.501±0.008	0.386±0.011	0.315±0.005	0.282±0.011
	MVGRL	**0.740±0.007**	0.542±0.010	0.505±0.007	0.387±0.008	0.311±0.007	0.295±0.009
	MERIT	0.730±0.005	0.544±0.010	0.503±0.008	0.388±0.009	0.314±0.011	0.300±0.009
	BRGCL	0.722±0.004	**0.562±0.007**	**0.514±0.012**	**0.399±0.012**	**0.331±0.012**	**0.312±0.010**
Coauthor CS	GCN *	0.918±0.001	0.702±0.010	0.628±0.012	0.531±0.010	0.455±0.011	0.415±0.013
	S^2GC *	**0.928±0.001**	0.713±0.010	0.638±0.010	0.556±0.009	0.476±0.012	0.422±0.012
	NRGNN	0.919±0.002	0.710±0.012	0.632±0.013	0.560±0.008	0.469±0.011	0.423±0.012
	SUGRL	0.913±0.001	0.706±0.008	0.633±0.008	0.561±0.008	0.465±0.009	0.412±0.008
	MVGRL	0.924±0.004	0.709±0.005	0.634±0.007	0.562±0.011	0.466±0.005	0.426±0.005
	MERIT	0.922±0.005	0.714±0.006	0.639±0.009	0.561±0.007	0.471±0.007	0.429±0.008
	BRGCL	0.920±0.003	**0.722±0.009**	**0.653±0.011**	**0.575±0.013**	**0.488±0.010**	**0.442±0.012**

B.2 STATISTICAL SIGNIFICANCE OF IMPROVEMENTS FOR NODE CLASSIFICATION WITH LABEL NOISE.

To validate the statistical significance of the improvements of BRGCL over competing methods, we further calculate the p-values of t-test between BRGCL and the second best baseline for each noise level and dataset for symmetric label noise, asymmetric label noise, and attribute label noise. The p-values for node classification with label noise are shown in Table 7. The p-values for node classification with attribute noise are shown in Table 8. We run all the experiments for 10 times with random initialization and injected symmetric and asymmetric label noise. The p-values for all datasets with all noise levels are less than 0.05, suggesting the statistically significant improvement of BRGCL over baseline methods.

Table 7: p-values of the t-test between BRGCL and the second best baseline on semi-supervised node classification with symmetric label noise and asymmetric label noise.

Dataset	Noise Level		
	40	60	80
	Asymmetric / Symmetric	Asymmetric / Symmetric	Asymmetric / Symmetric
Cora	0.0285 / 0.0021	0.0091 / 0.0038	0.0079 / 0.0217
Citeseer	0.0133 / 0.0024	0.0003 / 0.0013	0.0009 / 0.0401
PubMed	0.0051 / 0.0354	0.0219 / 0.0129	0.0279 / 0.0106
Coauthor CS	0.0341 / 0.0102	0.0121 / 0.0267	0.0317 / 0.0097
ogbn-arxiv	0.0393 / 0.0076	0.0039 / 0.0331	0.0095 / 0.0292

Table 8: p-values of the t-test between BRGCL and the second best baseline on semi-supervised node classification with attribute noise.

Dataset	Noise Level				
	40	50	60	70	80
Cora	0.0082	0.0193	0.0110	0.0267	0.0372
Citeseer	0.0019	0.0410	0.0394	0.0106	0.0289
PubMed	0.0286	0.0301	0.0371	0.0097	0.0165
Coauthor CS	0.0402	0.0122	0.0398	0.0051	0.0176
ogbn-arxiv	0.0219	0.0284	0.0314	0.0177	0.0120

B.3 NODE CLUSTERING WITH INPUT ATTRIBUTE NOISE

To further validate the performance of the node representation learned by BRGCL, we perform node clustering on clean benchmark datasets. We follow the same evaluation protocol as that in Section 5.2. K-means is then applied to the learned node representations to obtain the clustering results. It can be observed from Table 9 that BRGCL still outperforms all baseline methods for node clustering.

Table 9: Node clustering performance comparison on clean benchmark datasets.

Methods	Cora		Citeseer		PubMed		Coauthor CS		ogbn-arxiv	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Supervised										
GCN	68.3±0.71	52.3±0.54	68.8±0.65	41.9±0.24	69.1±0.99	31.2±0.46	69.8±0.34	68.6±0.59	52.0±1.02	68.0±0.74
S^2GC	69.6±0.42	54.7±0.65	69.1±0.82	42.8±0.55	70.1±0.89	33.2±0.31	70.2±0.45	67.0±0.72	53.6±0.79	68.5±0.70
NRGNN	72.1±0.53	55.6±0.49	69.3±0.77	43.6±0.51	69.9±1.03	34.2±0.59	68.8±0.59	66.2±0.84	51.9±0.84	68.1±0.65
Unsupervised										
K-means	49.2±0.56	32.1±0.53	54.0±0.43	30.5±1.03	59.5±0.67	31.5±0.77	35.2±0.76	20.1±0.92	31.6±0.75	35.9±0.96
GAE	59.6±0.67	42.9±0.62	57.9±0.38	17.6±1.01	65.2±0.37	27.7±0.35	46.7±0.88	41.6±0.81	38.9±0.91	48.4±1.02
ARVGA	64.0±0.41	45.0±0.59	57.3±0.51	26.1±0.54	69.0±0.60	29.0±0.44	60.3±0.61	55.9±0.65	49.2±0.80	68.3±0.65
GALA	74.5±0.57	57.6±0.68	69.3±0.60	44.1±0.39	69.3±0.58	32.7±0.42	66.5±0.79	68.8±0.48	53.5±0.65	66.5±0.72
GraphCL	71.9±0.66	54.6±0.59	68.3±0.42	42.7±0.63	67.6±0.42	31.5±0.32	65.2±0.53	66.4±0.69	52.1±0.67	67.6±0.63
MVGRL	74.2±0.54	57.3±0.44	69.6±0.31	44.7±0.56	69.6±0.44	33.9±0.46	69.3±0.61	69.2±0.57	53.2±0.61	68.2±0.50
MERIT	74.1±0.67	57.6±0.55	69.7±0.55	45.1±0.41	69.8±0.30	33.7±0.40	69.8±0.60	69.5±0.59	54.3±0.54	68.5±0.75
BRGCL	**74.8±0.39**	**58.1±0.51**	**70.3±0.39**	**45.7±0.63**	**70.4±0.39**	**34.9±0.45**	**71.4±0.43**	**70.2±0.56**	**55.5±0.52**	**70.2±0.71**
p-value	0.0009	0.0254	0.0439	0.0395	0.0429	0.0189	0.0099	0.0302	0.0189	0.0151

B.4 COMPARISONS TO EXISTING SAMPLE SELECTION METHODS

In this subsection, we compare BRGCL against previous sample selection methods, including Co-teaching (Han et al., 2018) and Self-Training (Li et al., 2018) for node classification with symmetric

label noise. Co-teaching maintains two networks to select clean samples for each other. Self-Training finds nodes with the most confident pseudo labels, and it augmented the labeled training data by incorporating confident nodes with their pseudo labels into the existing training data.The results are shown in Table 10. We can clearly see that BRGCL greatly outperforms competing sample selection methods.

Table 10: Performance comparison against Co-teaching (Han et al., 2018) and Self-training (Li et al., 2018) on node classification with different levels of symmetric label noise.

Dataset	Methods	Noise Level				
		40	50	60	70	80
Cora	Self-training	0.664±0.012	0.584±0.007	0.532±0.013	0.459±0.011	0.368±0.012
	Co-teaching	0.668±0.011	0.593±0.011	0.527±0.010	0.465±0.010	0.367±0.017
	BRGCL	**0.694±0.007**	**0.622±0.009**	**0.567±0.013**	**0.500±0.014**	**0.394±0.012**
Citeseer	Self-training	0.541±0.014	0.465±0.013	0.397±0.013	0.347±0.016	0.301±0.022
	Co-teaching	0.522±0.018	0.461±0.011	0.383±0.011	0.338±0.014	0.299±0.020
	BRGCL	**0.569±0.013**	**0.496±0.011**	**0.433±0.014**	**0.368±0.013**	**0.321±0.014**
PubMed	Self-training	0.597±0.019	0.507±0.011	0.419±0.021	0.380±0.020	0.345±0.023
	Co-teaching	0.584±0.013	0.499±0.015	0.403±0.014	0.371±0.011	0.342±0.022
	BRGCL	**0.632±0.010**	**0.530±0.010**	**0.468±0.010**	**0.399±0.012**	**0.349±0.013**
Coauthor CS	Self-training	0.672±0.010	0.614±0.012	0.542±0.013	0.462±0.015	0.397±0.015
	Co-teaching	0.666±0.012	0.610±0.011	0.529±0.015	0.451±0.013	0.404±0.019
	BRGCL	**0.710±0.008**	**0.638±0.009**	**0.572±0.011**	**0.480±0.011**	**0.428±0.015**
ogbn-arxiv	Self-training	0.462±0.012	0.413±0.014	0.368±0.018	0.328±0.014	0.276±0.020
	Co-teaching	0.437±0.024	0.406±0.011	0.359±0.016	0.322±0.012	0.282±0.025
	BRGCL	**0.482±0.006**	**0.432±0.009**	**0.399±0.009**	**0.344±0.012**	**0.296±0.013**

B.5 Joint Training vs. Decoupled Training.

We study the effectiveness of our decoupled training framework compared with jointly training the encoder and the classifier. We compare the performance on node classification with 20% label noise level. The results are shown in Table 11. It can be observed that decoupling the training of classifier and encoder can mitigate the effects of label noise.

Table 11: Ablation study on contrastive components for node classification with label noise.

Method	Cora		Citeseer		PubMed	
	Confident	K-means	Confident	K-means	Confident	K-means
Joint	77.7±0.08	78.2±0.11	65.7±0.09	65.3±0.10	72.5±0.10	72.5±0.12
Decoupled	79.3±0.09	78.5±0.09	66.8±0.07	66.3±0.08	73.4±0.07	73.0±0.09

B.6 Number of Confident Prototypes.

To further study the behavior of BRGCL, we show the number of robust prototypes estimated by BPL in Table 12. It can be observed from the results that the estimated number of robust prototypes is usually very close to the ground truth number of classes for different datasets, justifying the effectiveness of BPL. Because BEC is based on the pseudo labels estimated by BPL, the success of BPL leads to trustworthy estimation of confident nodes and robust prototypes by BEC.

Table 12: Number of robust prototypes inferred by BPL compared with the ground truth number of classes on different datasets

Datasets	Citeseer	Cora	PubMed	Coauthor CS	ogbn-arxiv
ξ in eq. (2)	0.15	0.20	0.35	0.30	0.4
Estimated K	6	8	3	19	48
Classes	6	7	3	15	40