Relational Self-Supervised Learning on Graphs

Namkyeong Lee
KAIST ISysE
Daejeon, Republic of Korea
namkyeong96@kaist.ac.kr

Dongmin Hyun
POSTECH PIAI
Pohang, Republic of Korea
dm.hyun@postech.ac.kr

Junseok Lee
KAIST ISysE
Daejeon, Republic of Korea
junseoklee@kaist.ac.kr

Chanyoung Park*
KAIST ISysE & AI
Daejeon, Republic of Korea
cy.park@kaist.ac.kr

ABSTRACT

Over the past few years, graph representation learning (GRL) has been a powerful strategy for analyzing graph-structured data. Recently, GRL methods have shown promising results by adopting self-supervised learning methods developed for learning representations of images. Despite their success, existing GRL methods tend to overlook an inherent distinction between images and graphs, i.e., images are assumed to be independently and identically distributed, whereas graphs exhibit relational information among data instances, i.e., nodes. To fully benefit from the relational information inherent in the graph-structured data, we propose a novel GRL method, called RGRL, that learns from the relational information generated from the graph itself. RGRL learns node representations such that the relationship among nodes is invariant to augmentations, i.e., *augmentation-invariant relationship*, which allows the node representations to vary as long as the relationship among the nodes is preserved. By considering the relationship among nodes in both global and local perspectives, RGRL overcomes limitations of previous contrastive and non-contrastive methods, and achieves the best of both worlds. Extensive experiments on fourteen benchmark datasets over various downstream tasks demonstrate the superiority of RGRL over state-of-the-art baselines. The source code for RGRL is available at https://github.com/Namkyeong/RGRL.

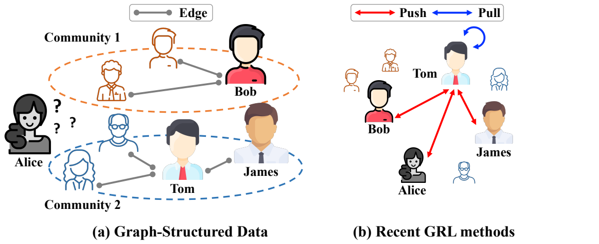
CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**; **Unsupervised learning**.

KEYWORDS

Self-Supervised Learning, Graph Representation Learning, Graph Neural Networks

ACM Reference Format:
Namkyeong Lee, Dongmin Hyun, Junseok Lee, and Chanyoung Park. 2022. Relational Self-Supervised Learning on Graphs. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557428



(a) Graph-Structured Data (b) Recent GRL methods

Figure 1: Recent GRL methods cannot fully benefit from the relational information of given graph-structured data.

1 INTRODUCTION

Recently, self-supervised learning paradigm, which trains models on pretext tasks derived solely from data without any label information, achieved great success in many domains [2, 3, 5, 6]. Among various self-supervised learning approaches, contrastive learning has shown its effectiveness in representation learning by pulling semantically similar (i.e., positive) pairs of data instances together and pushing dissimilar (i.e., negative) ones apart [3, 10]. In general, successful contrastive methods follow the principle of instance discrimination [35], which pairs data instances based on whether they are derived from the same instance (i.e., positive pairs) or not (i.e., negative pairs).

Inspired by the success of the contrastive methods in computer vision, these methods have been recently adopted to representation learning on graphs. However, existing contrastive learning-based GRL methods [32, 40, 41] closely follow the model architectures that were successful on images without considering an inherent distinction between images and graphs, i.e., images are assumed to be independently and identically distributed, whereas *graphs exhibit relational information among nodes*. For example, GRACE [40] and GCA [41] inherit the instance discrimination principle [3, 35], and treat all other nodes apart from the node itself as negatives. However, we argue that without considering the relational information inherent in graphs, these methods are prone to *sampling bias* [4, 17, 18], i.e., some negative samples are in fact semantically similar to the query node.

To illustrate the sampling bias of existing methods, consider Tom as the query node in Fig. 1. Then, James would be regarded as a negative sample (Fig. 1b) even though James belongs to the same community as Tom (Fig. 1a), which means that they are likely to share some interest. To make the matter worse, James would be treated equally as negative to Tom as Bob is to Tom (Fig. 1b), even though Bob belongs to a different community (Fig. 1a). These false supervisory signals can seriously interfere with representation learning on graphs, and eventually degrade the performance on downstream tasks, such as node classification, and link prediction. BGRL [30], which is a recent non-contrastive method, avoids the sampling bias by relying only on positive samples, i.e., "pull" only

[*] Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557428

in Fig. 1b. However, since BGRL is trained by predicting an augmented version of a node itself, it still cannot fully benefit from the relational information inherent in the graph-structured data. For example, assume that we lack information about Alice (i.e., less informative or noisy features) in Fig. 1a. In this case, if we only considered Alice herself along with her augmented version to train her own representation as done in BGRL (i.e., "pull" only in Fig. 1b), it would be non-trivial to discover which community she is likely to join. However, the problem can be alleviated by using relational information in the graph. That is, considering Alice's relationship with Bob and Tom would make it easier to discover which community Alice is likely to join.

To this end, we propose Relational Graph Representation Learning (RGRL), a simple yet effective self-supervised learning framework for graphs, that benefits from the relational information inherent in the graph-structured data. The main idea is to *allow node representations to vary as long as the relationship among the nodes is preserved*, rather than 1) strictly distinguishing positive nodes from negative nodes as done by existing contrastive methods [40, 41], or 2) strictly enforcing the node representations to be augmentation-invariant as done by existing non-contrastive methods [30, 37]. More precisely, given two GNN-based encoders each of which encodes an augmented view of a graph, the first encoder calculates the similarity of a query node with respect to a set of sampled anchor nodes, while the second encoder tries to mimic the computed query-anchors similarity. By doing so, RGRL 1) relaxes the binary nature of contrastive methods with soft labeling, thereby alleviating the sampling bias, and 2) learns node representations such that the relationship among nodes is invariant to augmentations, i.e., *augmentation-invariant relationship*, rather than augmentation-invariant node representation [30, 41]. Consequently, we expect RGRL to capture the core relationship among nodes that should be preserved no matter how the graph is perturbed.

A key challenge of RGRL is how to sample anchor nodes, as GNN-based models are known to be *degree-biased* [29], i.e., nodes with higher degree tend to result in higher quality representations, as shown in Fig. 2. This is caused by the fact that high-degree nodes receive more information during neighborhood aggregation compared with low-degree nodes. Hence, we introduce a technique to sample anchor nodes so that nodes with lower degree are sampled more frequently than nodes with higher degree. By doing so, the parameters of RGRL are optimized by focusing more on low-degree nodes, thereby alleviating the degree-biased issue. Moreover, besides sampling the anchor nodes in the *global* perspective, we additionally sample nodes that are structurally close to the query node to capture the *local* structural information given in a graph.

Our extensive experiments on **fourteen** real-world datasets on various downstream tasks, i.e., node classification in both homogeneous and heterogeneous graphs, and link prediction, demonstrate the superiority of RGRL. Moreover, our qualitative analysis shows that RGRL indeed captures the core relationship among nodes. Further appeals of RGRL are that RGRL learns high-quality representations of nodes with 1) less informative input features, which demonstrates the robustness of RGRL, and nodes with 2) low degree, both of which demonstrate the practicality of RGRL in real-world applications.

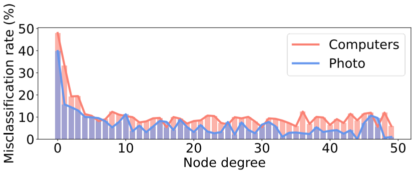


Figure 2: GNN-based models are biased towards high-degree nodes. Low-degree nodes tend to be misclassified.

2 RELATED WORK

2.1 Graph Representation Learning

Recent years have seen a surge of interest in analyzing graph-structured data by using various machine learning approaches. Inspired by word2vec [21], DeepWalk [25] and node2vec [7] generate node sequences through random walks, and apply skip-gram approach to map nodes appearing within the same context into similar vector representations. While these models have shown success in various tasks such as node classification and link prediction, they overemphasize the proximity information at the expense of structural information [32], and cannot incorporate node attributes [13, 32]. Graph Neural Networks (GNNs) [8, 13, 31] address these limitations by learning node representations through recursively aggregating the features of neighboring nodes. However, GNNs still require a sufficient number of labeled data for training, which is impractical in reality [32].

2.2 Self-Supervised Learning on Graphs

Contrastive learning-based. Motivated by the great success of contrastive methods in computer vision applied on images, these methods have recently been adopted to graphs [9, 16, 37]. Inspired by Deep Infomax [10], DGI [32] learns node representations by maximizing the mutual information between the local patch of a graph, i.e., node, and the global summary of the graph, thereby capturing the global information of a graph that is overlooked by vanilla graph convolutional networks (GCNs). GRACE [40], which adopts SimCLR [3] to graph domain, learns node representations with two augmented views of a graph. Specifically, GRACE first creates two augmented views of a graph by randomly dropping edges or masking their features. Then, it pulls representations of the same node in the two augmented graphs while pushing apart representations of every other node. GCA [41] enhances GRACE by introducing adaptive augmentation techniques that focus on the graph structure. Despite the success of contrastive methods on graphs, they suffer from the sampling bias issue incurred by false negatives among the negative samples [4, 18, 36]. Although recent DGCL [36] alleviates the issue by utilizing the probability as the weight of negative samples, it still treats all other nodes as negative samples, thus cannot fully leverage the relational information inherent in graphs. Moreover, these methods require high computational cost and memory usage owing to a large amount of negative samples required for the model training [30].

Non-contrastive learning-based. Recent non-contrastive methods avoid the aforementioned limitations by not using negative samples [30, 37]. BGRL [30] learns node representations by encoding two augmented versions of a graph using two separate encoders:

one is trained by maximizing the cosine similarity between the representations generated by the two encoders, while the other encoder is updated by an exponential moving average of the first encoder. While following the conventional augmentation-invariant learning scheme, CCA-SSG [37] learns node representations by incorporating additional decorrelation terms so that each dimension of the node representation captures distinct semantics of the node. Despite avoiding the sampling bias, non-contrastive learning methods still overlook the relational information among nodes in a graph by treating each node to be independent from other nodes after message passing. In this work, we propose a general framework for learning node representations that leverages relational information as supervisory signals to learn augmentation-invariant relationship among nodes.

3 PROBLEM STATEMENT

Notations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, ..., v_N\}$ represents the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. $\mathcal{G}$ is associated with a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ where $\mathbf{A}_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ otherwise.
Task: Unsupervised Graph Representation Learning. Given a graph $\mathcal{G}$ along with $\mathbf{X}$ and $\mathbf{A}$, we aim to learn an encoder $f(\cdot)$ that produces node representations $\mathbf{H} = f(\mathbf{X}, \mathbf{A}) \in \mathbb{R}^{N \times D}$, where $D \ll F$. Our goal is to learn node representations that generalize well to various downstream tasks without using any labeled data.

4 PROPOSED FRAMEWORK: RGRL

In this section, we first explain how RGRL learns the representation of a query node by preserving its similarity with anchor nodes (**Sec. 4.1**) that are sampled regarding both global (**Sec. 4.2.1**) and local (**Sec. 4.2.2**) perspectives. Then, we explain how the parameters of RGRL are updated (**Sec. 4.3**). Lastly, we introduce how RGRL can be extended to heterogeneous graphs **(Sec. 4.4)**. Fig. 3 illustrates the overall architecture of RGRL[1].

4.1 Relational Graph Representation Learning

In contrast to existing contrastive methods on graphs, RGRL focuses on preserving the relationship among nodes for learning node representations. More precisely, we first generate two graph views $\tilde{\mathcal{G}}_1 = (\tilde{\mathbf{X}}_1, \tilde{\mathbf{A}}_1)$ and $\tilde{\mathcal{G}}_2 = (\tilde{\mathbf{X}}_2, \tilde{\mathbf{A}}_2)$ by applying a stochastic graph augmentation function $\mathcal{T}_1$ and $\mathcal{T}_2$ to the original graph $\mathcal{G}$, respectively. Then, the online encoder f_θ produces online representation $\tilde{\mathbf{H}}^\theta = f_\theta(\tilde{\mathbf{X}}_1, \tilde{\mathbf{A}}_1) \in \mathbb{R}^{N \times D}$, while the target encoder f_ξ produces target representation $\tilde{\mathbf{H}}^\xi = f_\xi(\tilde{\mathbf{X}}_2, \tilde{\mathbf{A}}_2) \in \mathbb{R}^{N \times D}$. The online representation $\tilde{\mathbf{H}}^\theta$ is additionally fed into a node-level predictor g_θ to obtain the prediction of the target representation, i.e., $\tilde{\mathbf{Z}}^\theta = g_\theta(\tilde{\mathbf{H}}^\theta) \in \mathbb{R}^{N \times D}$.

Given a query node $v_i \in \mathcal{V}$, we first sample a set of anchor nodes $\mathbf{N}_i$ from graph $\mathcal{G}$. Then, the similarity is calculated between the target embedding $\tilde{\mathbf{h}}_i^\xi$ of the query node v_i, and the target embeddings $\tilde{\mathbf{h}}_j^\xi$ of anchor nodes $v_j \in \mathbf{N}_i$, divided by a temperature hyperparameter τ_ξ. The similarity is then converted into a probability distribution via softmax :

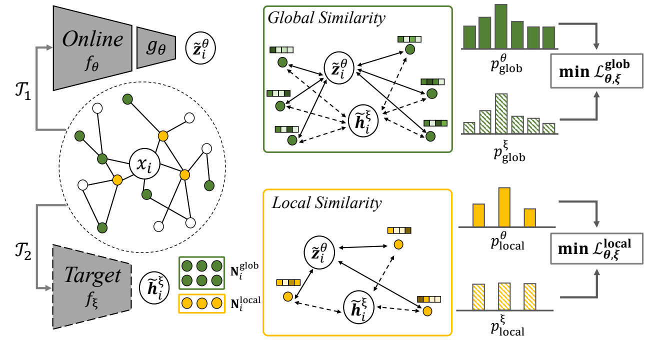

Online
f_θ
g_θ
$\tilde{\mathbf{z}}_i^\theta$
$\mathcal{T}_1$
x_i
$\mathcal{T}_2$
Target
f_ξ
$\tilde{\mathbf{h}}_i^\xi$
$\mathbf{N}_i^{\text{glob}}$
$\mathbf{N}_i^{\text{local}}$
Global Similarity
$\tilde{\mathbf{z}}_i^\theta$
$\tilde{\mathbf{h}}_i^\xi$
p_{glob}^θ
p_{glob}^ξ
min $\mathcal{L}_{\theta,\xi}^{\text{glob}}$
Local Similarity
$\tilde{\mathbf{z}}_i^\theta$
$\tilde{\mathbf{h}}_i^\xi$
p_{local}^θ
p_{local}^ξ
min $\mathcal{L}_{\theta,\xi}^{\text{local}}$

Figure 3: Overall model architecture of RGRL.

$$p_i^\xi(j) = \frac{\exp(\mathtt{sim}(\tilde{\mathbf{h}}_i^\xi, \tilde{\mathbf{h}}_j^\xi)/\tau_\xi)}{\sum_{k \in \mathbf{N}_i} \exp(\mathtt{sim}(\tilde{\mathbf{h}}_i^\xi, \tilde{\mathbf{h}}_k^\xi)/\tau_\xi)}, \forall v_j \in \mathbf{N}_i \quad (1)$$

where $p_i^\xi(j) \in \mathbb{R}$ is the j-th element of $p_i^\xi \in \mathbb{R}^{|\mathbf{N}_i|}$, and $\mathtt{sim}(\cdot,\cdot)$ denotes the cosine similarity between the two input representations.

Likewise, we calculate the similarity between the online prediction $\tilde{\mathbf{z}}_i^\theta$ of the query node v_i, and the target embeddings $\tilde{\mathbf{h}}_j^\xi$ of anchor nodes $v_j \in \mathbf{N}_i$ as follows:

$$p_i^\theta(j) = \frac{\exp(\mathtt{sim}(\tilde{\mathbf{z}}_i^\theta, \tilde{\mathbf{h}}_j^\xi)/\tau_\theta)}{\sum_{k \in \mathbf{N}_i} \exp(\mathtt{sim}(\tilde{\mathbf{z}}_i^\theta, \tilde{\mathbf{h}}_k^\xi)/\tau_\theta)}, \forall v_j \in \mathbf{N}_i \quad (2)$$

where $p_i^\theta(j)$ is the j-th element of $p_i^\theta \in \mathbb{R}^{|\mathbf{N}_i|}$, and τ_θ is the temperature hyperparameter of online network.

Having computed two probability distributions for each node $v_i \in \mathcal{V}$, i.e., p_i^θ and p_i^ξ, derived from the query-anchors similarity, we minimize the following sum of KL divergence:

$$\mathcal{L}_{\theta,\xi} = \sum_{v_i \in \mathcal{V}} KL(p_i^\theta \,||\, p_i^\xi). \quad (3)$$

By minimizing the above loss, the online network is trained to mimic the relational information captured by the target encoder, thereby learning node representations such that the relationship among nodes is invariant to augmentations, i.e., *augmentation-invariant relationship*, which is the core relationship among nodes that should be preserved no matter how the graph is perturbed. Note that the role of the temperature hyperparameters, i.e., τ_ξ and τ_θ, is to control the discriminability among the sampled anchor nodes in the representation space by determining the sharpness of the distribution [39]. That is, given that τ_θ is fixed, as τ_ξ becomes smaller, the target distribution becomes more sharpened, providing a discriminative guidance to the online network.

4.2 Sampling Anchor Nodes

A key challenge of RGRL is how to sample anchor nodes for each node $v_i \in \mathcal{V}$, i.e., $\mathbf{N}_i$. We argue that diverse relational information regarding both global and local perspectives should be considered. Thus, we sample two distinct sets of anchor nodes $\mathbf{N}_i^{\text{glob}}$ **(Sec. 4.2.1)**

[1] RGRL adopts BYOL [6] as the backbone of the framework, which is a recently proposed non-contrastive method for image representation learning.

and $\mathbf{N}_i^{\text{local}}$ (**Sec. 4.2.2**), that play different roles. Specifically, the global anchor nodes $\mathbf{N}_i^{\text{glob}}$ are utilized to learn diverse relationship among the nodes considering the overall graph structure, while the local anchor nodes $\mathbf{N}_i^{\text{local}}$ are leveraged to learn the local fine-grained relationship among the nodes that are structurally close in the graph.

4.2.1 ***Capturing global similarity.*** To capture the similarity among the nodes in the global perspective, we can naïvely sample anchor nodes uniformly from the entire graph. However, such a uniform sampling strategy overlooks the highly-skewed node degree distribution, which is not desired since the quality of node representations is closely related to the node degree. To verify this, we conduct an empirical analysis on two real-world graphs, i.e., Amazon Computers and Amazon Photo [19], each of which exhibits a power law distribution of node degree. We perform node classification on node representations trained with the state-of-the-art self-supervised method, i.e., BGRL [30]. As shown in Fig. 2, we observe that the misclassification rate of low-degree nodes is significantly higher than that of high-degree nodes indicating that the training is biased towards high-degree nodes. We attribute this to the neighborhood aggregation scheme of GNNs in that low-degree nodes receive less information compared with high-degree nodes, which leads to underfitting of GNNs to low-degree nodes. It is worth noting that our observation aligns with findings of [29] whose experiments are done under the semi-supervised setting, demonstrating that the *degree-bias* issue is critical for the model performance regardless of the existence of the label information. To the best of our knowledge, RGRL is the first work that addresses the degree-bias issue in self-supervised learning on graphs.

Based on the above observation, we propose to focus on low-degree nodes while training RGRL. The key idea is to sample a set of anchor nodes v_j from *inverse degree-weighted distribution*, which is designed to sample low-degree nodes more frequently as follows:

$$w_j = \alpha^{\log(\deg_j + 1)} + \beta \tag{4}$$

where $\deg_j$ is the degree of node v_j, α and β are hyperparameters controlling the skewness of the distribution and the minimum sampling weight, respectively. It is worth noting that α is set to a value between 0 and 1 to *approximate the misclassification rate distribution shown in Fig. 2.* This implies that nodes with high misclassification rate will be sampled more frequently thereby alleviating the degree-based issue of GNNs. Finally, we normalize the sampling weight across the nodes, and assign the sampling probability of node v_j as:

$$p_{sample}(j) = \frac{w_j}{\sum_{v_k \in \mathcal{V}} w_k}, \forall v_j \in \mathcal{V} \tag{5}$$

In summary, given a query node v_i, we sample a set of anchor nodes $\mathbf{N}_i^{\text{glob}}$ from the distribution defined in Eqn. 5. Then, we compute the two similarity distributions defined in Eqn. 1 and Eqn. 2, i.e., p_{glob}^{θ} and p_{glob}^{ξ} followed by their KD divergence loss, i.e., $\mathcal{L}_{\theta,\xi}^{\text{glob}}$, defined in Eqn. 3. It is worth noting that the similarity distributions computed in this manner capture the query-anchors similarity in the *global perspective*, since the anchor nodes are sampled from the entire graph regardless of the structural information of the graph.

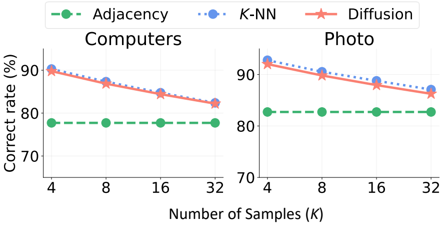


Figure 4: Analysis on the ratio of its neighboring nodes being the same label as the query node across different Ks.

4.2.2 ***Capturing local similarity.*** By capturing the similarity among nodes in the global perspective, RGRL learns diverse relationship among nodes without accounting for the explicit structural information, which is crucial for learning local relationship in graph-structured data. Although the GNN encoders naturally capture the local structural information through the neighborhood aggregation scheme, we argue that GNNs fail to capture fine-grained relationship among nodes, i.e., the relationship among the nodes that belong to the same class. For example, in an academic publication network, even though authors who belong to "Data Mining" class and "Machine Learning" class may be structurally close to each other due to their similarity in some research topics, these classes are distinguished from each other in a fine-grained perspective. In other words, authors who belong to "Data Mining" class are more similar to each other than to authors who belong to "Machine Learning". However, since GNNs map nodes that are structurally close to similar representations, authors from the two different classes may be mapped to similar representations. Hence, to capture fine-grained relationship among structurally close nodes, we propose to sample anchor nodes that 1) are structurally close with the query node in the graph, and at the same time 2) share the same label information with the query node. Such a set of anchor nodes can be discovered by various approaches, such as adjacency, K-NN and diffusion:

1) Adjacency: The most naïve approach is to consider the neighboring nodes of a query node as the set of local anchor nodes. Although the adjacency matrix contains local connectivity information among nodes, not all the adjacent relationship reveals identical semantics among nodes. As shown in Fig. 4, the pure adjacency information contains many false positive relationships among the nodes, and thus it is not appropriate as $\mathbf{N}_i^{\text{local}}$. For example, more than 20% and 15% of neighboring nodes of a query node have different labels in Computers and Photo datasets on average, respectively. **2) K-NN:** Instead of relying on the adjacency information, we can use K-NN approach to sample K nodes that are most similar to the query node according to the learned node representations. Although we can greatly reduce the number of false positives as shown in Fig. 4, K-NN not only discovers nodes that are explicitly connected to the query node but also those that share similar features, i.e., distant but similar nodes, which contradicts with our intention of capturing the similarity among nodes in the local perspective. Moreover, it requires additional $O(|\mathcal{V}|^2)$ space and time complexity, which makes it impractical to apply RGRL on real-world large graphs. **3) Diffusion:** To this end, diffusion-based approach [14] can be considered as a compromise between the above approaches. Graph diffusion calculates the closeness of nodes in the graph structure by repeatedly passing the weighting coefficients to the neighboring nodes. As shown in Fig. 4, anchor nodes sampled with high diffusion scores have high probability of sharing the same label as much as K-NN even though diffusion only considers the structural information. Furthermore, diffusion does not require any additional computation during training since diffusion scores can be readily computed before the model training. Hence, in this paper, we leverage diffusion to sample local anchor nodes, i.e., $\mathbf{N}_i^{\text{local}}$.

More precisely, we calculate a diffusion matrix [14] $\mathbf{S}$, based on personalized PageRank (PPR) [22] as follows:

$$\mathbf{S} = \sum_{k=0}^{\infty} t(1-t)^k \mathbf{T}^k \tag{6}$$

where $t \in (0,1)$ is the teleport probability, and $\mathbf{T}$ is the symmetric transition matrix $\mathbf{T} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal matrix of node degrees, i.e, $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ij}$. Each component (i,j) of the diffusion matrix $\mathbf{S}$ indicates the closeness of node v_i and v_j. For a query node v_i, we pre-define the top-K highest scoring nodes denoted by $\mathbf{N}_i^{local}$ before the training of RGRL. During the training, we calculate the KL divergence loss defined in Eqn. 3, i.e., $\mathcal{L}_{\theta,\xi}^{\text{local}}$, between the similarity distributions $p_{\text{local}}^{\theta}$ and p_{local}^{ξ} computed based on $\mathbf{N}_i^{\text{local}}$ through Eqn. 1 and Eqn. 2, respectively.

4.3 Model update

4.3.1 *Updating online encoder f_θ and predictor g_θ.*

During the training, the online parameters θ are updated to jointly minimize both global and local losses, i.e., $\mathcal{L}_{\theta,\xi}^{\text{glob}}$ and $\mathcal{L}_{\theta,\xi}^{\text{local}}$, as follows: $\mathcal{L}_{\theta,\xi} = \mathcal{L}_{\theta,\xi}^{\text{glob}} + \lambda \cdot \mathcal{L}_{\theta,\xi}^{\text{local}}, \quad \theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}_{\theta,\xi}, \eta\right)$, where λ controls the importance of local structural information, and η is the learning rate for online network.

4.3.2 *Updating target encoder f_ξ.*

RGRL updates the target encoder by smoothing the parameter of online encoder with the decay rate γ: $\xi \leftarrow \gamma\xi + (1-\gamma)\theta$.

4.3.3 *Graph augmentation functions.*

Following previous work [30, 40], we use standard graph perturbations for augmentation functions $\mathcal{T}_1$ and $\mathcal{T}_2$. Specifically, we randomly mask node features and drop edges with fixed probabilities ($p_{f,1}$ and $p_{e,1}$ for $\mathcal{T}_1$, and $p_{f,2}$ and $p_{e,2}$ for $\mathcal{T}_2$). We consider only simple and standard augmentation functions to study the effect of RGRL as a representation learning framework.

4.4 Extension to Heterogeneous Graphs

Thanks to the generality of RGRL as a relational learning framework, it can be naturally extended to an attributed multiplex network [12, 23, 34], which is a type of heterogeneous graph [27]. As a multiplex network consists of multiple layers of attributed graphs, we consider each layer of the multiplex network as a view instead of generating multiple views through augmentations. Then, we make pairs with the layers to perform RGRL. For example, given four layers of attributed graphs, we make six pairs of graphs in total, i.e., $\binom{4}{2} = 6$. At inference time, we employ a mean pooling function to aggregate layer-specific node representations.

5 DISCUSSION

In this section, we address limitations of previous state-of-the-art methods of two different types, i.e., contrastive learning (GRACE [40] /GCA [41]) and non-contrastive learning (BGRL [30]), and discuss how RGRL overcomes their limitations by leveraging relational information among nodes.

Comparison to contrastive learning-based methods. Under the principle of instance discrimination [35], GRACE/GCA learns node representations by contrasting the representation of a query node with the representations of all other nodes in the two augmented graphs, while matching the representations of the same node. That is, GRACE/GCA simply treats all other nodes apart from the query node itself as negatives even if they may share similar semantics with the query node, i.e., sampling bias [4, 18]. What's even worse here is that the softmax-based contrastive loss used in GRACE/GCA is a *hardness-aware loss* function [33], which gives larger penalties to the anchor nodes that are closer (or more similar) to the query node. Such a hardness-aware loss would facilitate the model to learn more discriminative boundary in the supervised setting where negative samples can be readily obtained. However, this is not the case in self-supervised learning methods. As shown in Fig 4, a large proportion of nearest-neighbors of a query node obtained by K-NN approach shares the same label as the query node. As all other nodes apart from the query node are treated as negatives, these close anchor nodes that share the same label as query node will not only be treated as negatives, but also be given more penalties as they are close to the query node, i.e., these false negative nodes will be trained to be even more dissimilar to the query node compared with other true negative nodes [33]. These false supervisory signals would seriously interfere with the representation learning process. On the other hand, RGRL relaxes the strict binary classification of GRACE/GCA with soft labeling so that the model can decide how much to push or pull other nodes based on the relational information among the nodes without relying on the binary decisions of positives and negatives.

Comparison to non-contrastive learning-based method. BGRL learns node representations by constraining the representations from two encoders to be close to each other without any use of negative samples, which addresses the aforementioned limitations of existing contrastive learning-based methods. However, as will be later shown in Fig. 5, we observe that BGRL suffers from a severe performance degradation when the feature information is not fully informative, i.e., contains noise, which is very common in reality. We attribute such behavior of BGRL to the strict self-preserving loss, which considers each node to be independent from other nodes after message passing, and enforces each node to be augmentation-invariant. Due to the strict constraint, BGRL may overfit to a few non-informative features leading to a severe performance degradation when the features are noisy. However, we argue that the overfitting problem can be overcome with a little help from other nodes in the graph, i.e., learning from the relationship with other nodes. To this end, RGRL relaxes the strict self-preserving loss with relation-preserving loss, allowing the representations to vary as long as the relationship among the representations is preserved. Thanks to the flexibility of the relation-preserving loss, RGRL is robust to the quality of node features as will be shown in Fig. 5.

Table 1: Comparison on computational complexity

Model	Complexity
GRACE	$4C_{\text{encoder}}(M+N) + 4C_{\text{projection}}N + C_{\text{GRACE}}(N^2)$
BGRL	$6C_{\text{encoder}}(M+N) + 4C_{\text{prediction}}N + C_{\text{BGRL}}(N)$
RGRL	$6C_{\text{encoder}}(M+N) + 4C_{\text{prediction}}N + C_{\text{RGRL}}(NK)$

Computational Complexity Analysis. Assume we are given a graph with N nodes and M edges, and encoder f that computes embeddings in time and space complexity of $O(N+M)$. RGRL and BGRL [30] perform four encoder computations per update step with an additional node-level prediction step, while GRACE [40] performs two encoder computations with a node-level projection step. Assuming that a backpropagation is approximately as costly as a forward pass, the total time and space complexities per update step are given in Table 1, where C are constants depending on the architecture of the different components, and K is the number of samples in RGRL, i.e., the size of the set of anchor nodes $\mathbf{N}_i$ for each node v_i. As shown in Table 1, the complexity of RGRL increases linearly with K. However, as shown in Section 6.4, RGRL is robust over different Ks, and thus we set K to a value that is far smaller than the number of nodes N. In summary, since $K \ll N$, RGRL is more efficient than GRACE, and only entails slight increase of complexity compared with BGRL. Note that the node sampling probability and the diffusion matrix computation can be readily done before the model training, thus do not require any additional computational cost during the training process.

6 EXPERIMENTS

6.1 Experimental Setup

Datasets. We use **fourteen** widely used datasets to comprehensively evaluate the performance of RGRL on various downstream tasks, i.e., node classification and link prediction. The datasets include Wiki-CS [20], Amazon (*Computers* and *Photo*) [19], Coauthor (*Co.CS* and *Co.Physics*) [28], Plantoid (*Cora*, *Citeseer* and *Pubmed*), Cora Full [1], ogbn-arXiv [11], Reddit [8], and protein-protein interaction network (*PPI*) [8, 42]. Moreover, we use IMDB and DBLP [23] to evaluate RGRL on node classification on multiplex networks. The detailed statistics are summarized in Table 2.

Methods Compared. We compare RGRL with recent state-of-the-art graph representation learning methods, i.e., GRACE [40], GCA [41], CCA-SSG [37], and BGRL [30]. During the experiment, we use the official codes published by authors and then conduct evaluations within the same environment. For node classification, we also report previously published results of raw features (Feats.) and other representative methods, such as node2vec (n2v) [7], DeepWalk (DW) [25], DGI [32], GMI [24], and MVGRL [9] as done in [30, 41]. For evaluations on multiplex networks, we compare RGRL against recent state-of-the-art multiplex network representation learning methods, i.e., HAN [34], DMGI [23], and HDMI [12].

Table 2: Statistics for datasets used for experiments.

Dataset	Type	# Nodes	# Edges	# Features	# Cls.
WikiCS [2]	reference	11,701	216,123	300	10
Amazon Computers [3]	co-purchase	13,752	245,861	767	10
Amazon Photo [3]	co-purchase	7,650	119,081	745	8
Coauthor CS [3]	co-author	18,333	81,894	6,805	15
Coauthor Physics [3]	co-author	34,493	247,962	8,415	5
Cora [4]	citation	2,708	5,429	1,433	7
Citeseer [4]	citation	3,327	4,732	3,703	6
Pubmed [4]	citation	19,717	44,338	500	3
Cora Full [5]	citation	19,793	65,311	8,710	70
ogbn-arXiv [6]	citation	169,343	1,166,243	128	40
Reddit [7]	community	231,443	11,606,919	602	41
PPI (24 Graphs) [8]	interaction	56,944	818,716	50	121
IMDB [9]	co-actor co-director	3,550	66,428 13,788	2,000	3
DBLP [9]	co-author co-paper co-term	7,907	144,738 90,145 57,137,515	2,000	4

Evaluation Protocol. For node classification, we first train models in an unsupervised manner, and use the learned node representations to train and test a simple logistic regression classifier [32]. We use a random split of the nodes into train/validation/test nodes of 10/10/80%, respectively, except for WikiCS, ogbn-arXiv, Reddit and PPI datasets for which public splits are given. For link prediction, we first randomly split the original graph into train/validation/test edges of 50/20/30%, i.e., E_{train}, E_{val}, and E_{test}, and generate the same amount of negative edges, i.e., $\text{E}_{\text{train}_{\text{neg}}}$, $\text{E}_{\text{val}_{\text{neg}}}$, and $\text{E}_{\text{test}_{\text{neg}}}$, from the nodes that are not connected in the original graph [38]. We conduct experiments on two types of negative samples, i.e., random negatives that are sampled randomly among the pairs that are not directly connected in the original graphs, and hard negatives that are sampled among the pairs that are not directly connected but are located within three hop distances from the target node. After learning the node representations with E_{train}, we train a logistic regression classifier with E_{train} and $\text{E}_{\text{train}_{\text{neg}}}$, and test on E_{val}, $\text{E}_{\text{val}_{\text{neg}}}$, E_{test}, and $\text{E}_{\text{test}_{\text{neg}}}$. For both tasks, we report the test performance when the performance on validation set gives the best result. We measure performance in terms of Accuracy, Macro-f1 and Micro-f1 for node classification and Area Under Curve (AUC) and Average Precision (AP) for link prediction.

Implementation Details. Following previous works [30, 32, 40], we employ three distinct model architectures for various tasks (i.e. transductive learning, inductive learning on large graphs (Reddit dataset), and inductive learning on multiple graphs (PPI dataset)). We also follow the embedding dimensions, optimizer, activation function and augmentation hyperparameters of the state-of-the-art baseline, i.e., BGRL [30], while hyperparameters that are newly introduced for RGRL are tuned in certain ranges.

6.2 Performance Analysis

Evaluation on node classification. The empirical performance on node classification is summarized in Table 3. We have the following observations: **1)** Our relational learning framework RGRL outperforms all other baseline methods that overlook the relationship among nodes, i.e., GRACE, GCA, CCA-SSG, and BGRL. We argue that as RGRL allows the node representations to vary as long as the relationship with other nodes is preserved, the learned node representations capture the relational information among the nodes, thereby improving the performance of node classification. Considering that graphs reveal relationship among nodes, our relational learning framework aligns with the very nature of graph-structured

[2] https://github.com/pmernyei/wiki-cs-dataset/raw/master/dataset
[3] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/
[4] https://github.com/kimiyoung/planetoid/raw/master/data
[5] https://github.com/abojchevski/graph2gauss/raw/master/data/cora.npz
[6] http://snap.stanford.edu/ogb/data/nodeproppred/arxiv.zip
[7] https://data.dgl.ai/dataset/reddit.zip
[8] https://data.dgl.ai/dataset/ppi.zip
[9] https://www.dropbox.com/s/48oe7shjq0ih151/data.tar.gz?dl=0

Table 3: Performance on node classification tasks (OOM: Out of Memory on 24GB RTX3090).

	WikiCS	Computers	Photo	Co.CS	Co.Physics
GCN	77.19 (0.12)	86.51 (0.54)	92.42 (0.22)	93.03 (0.31)	95.65 (0.16)
Feats.	71.98 (0.00)	73.81 (0.00)	78.53 (0.00)	90.37 (0.00)	93.58 (0.00)
n2v	71.79 (0.05)	84.39 (0.08)	89.67 (0.12)	85.08 (0.03)	91.19 (0.04)
DW	74.35 (0.06)	85.68 (0.06)	89.44 (0.11)	84.61 (0.22)	91.77 (0.15)
DW+Feats.	77.21 (0.03)	86.28 (0.07)	90.05 (0.08)	87.70 (0.04)	94.90 (0.09)
DGI	75.35 (0.14)	83.95 (0.47)	91.61 (0.22)	92.15 (0.63)	94.51 (0.52)
GMI	74.85 (0.08)	82.21 (0.31)	90.68 (0.17)	OOM	OOM
MVGRL	77.52 (0.08)	87.52 (0.11)	91.74 (0.07)	92.11 (0.12)	95.33 (0.03)
GRACE	78.25 (0.65)	88.15 (0.43)	92.52 (0.32)	92.60 (0.11)	OOM
GCA	78.30 (0.62)	88.49 (0.51)	92.99 (0.27)	92.76 (0.16)	OOM
CCA-SSG	77.88 (0.41)	87.01 (0.41)	92.59 (0.25)	92.77 (0.17)	95.16 (0.10)
BGRL	79.60 (0.60)	89.23 (0.34)	93.06 (0.30)	92.90 (0.15)	95.43 (0.09)
RGRL	**80.29 (0.72)**	**89.70 (0.44)**	**93.43 (0.31)**	**92.94 (0.13)**	**95.46 (0.10)**

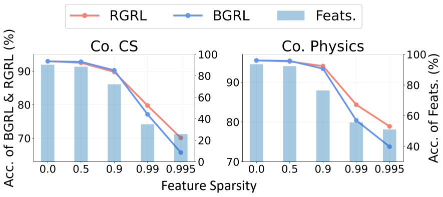


Figure 5: Node classification accuracy over feature sparsity. RGRL is robust to low-quality features.

data, which have been overlooked in previous methods. **2)** It is worth noting that methods built upon the instance discrimination principle [35], i.e., GRACE and GCA, not only are memory consuming (OOM on large datasets), but also generally perform worse than their counterpart, i.e., BGRL. This indicates that instance discrimination, which treats all other nodes except itself as negatives without considering the graph structural information, is not appropriate for graph-structured data that contain relational information among nodes. **3)** We find out that RGRL shows consistent improvements in datasets whose given feature is less informative (i.e., WikiCS, Computers, and Photo) as shown by the performance of "Feats." in Table 3. Thanks to the external self-supervisory signals from other nodes, RGRL performs well even without rich information of a single node. To corroborate the results, we conduct additional experiments by randomly corrupting a certain ratio of the input features of Co.CS and Co.Physics datasets whose features are relatively more informative as shown in Fig. 5. We indeed observe that RGRL is more robust than BGRL as the quality of the input features gets worse. We argue that this is mainly because RGRL learns the representation of a node from its relationship with other nodes in the graph rather than relying on the information contained in the node itself.

We also conduct experiments on commonly used small datasets (i.e., Cora, Citeseer, Pubmed, CoraFull), and a large dataset (i.e., ogbn-arXiv) as shown in Table 4. For ogbn-arXiv, we report results on both validation and test sets following [30], since the dataset is split according to the chronological order. **1)** As shown in Table 4, RGRL outperforms all other recent self-supervised learning methods on various datasets. **2)** Lastly, we conduct inductive node classification on Reddit and PPI datasets. We observe that RGRL performs the best, which demonstrates the inductive capability of RGRL. We attribute this to the flexibility of the relation-preserving loss of RGRL that helps avoid overfitting to the training data compared with the previous methods as discussed in Sec. 5, which eventually leads to improvements on unseen data.

Table 4: Performance on transductive node classification on other datasets (Accuracy), and inductive node classification on Reddit and PPI datasets (Micro-F1).

	Transductive						Inductive	
	Cora	Cite-seer	Pub-med	Cora Full	ogbn-arXiv		Reddit	PPI
					Valid	Test		
GRACE	83.38 (0.95)	70.79 (0.83)	83.96 (0.29)	64.19 (0.36)	OOM	OOM	94.84 (0.03)	67.12 (0.05)
GCA	82.79 (1.01)	70.70 (0.91)	84.19 (0.32)	64.34 (0.42)	OOM	OOM	94.85 (0.06)	66.72 (0.08)
CCA-SSG	83.01 (0.66)	70.35 (1.23)	84.81 (0.22)	64.09 (0.37)	59.43 (0.05)	58.50 (0.08)	94.89 (0.02)	66.09 (0.01)
BGRL	82.82 (0.86)	69.06 (0.80)	**86.16** **(0.19)**	63.94 (0.39)	70.66 (0.06)	69.61 (0.09)	94.90 (0.04)	68.89 (0.08)
RGRL	**83.98** **(0.78)**	**71.29** **(0.87)**	85.33 (0.20)	**64.62** **(0.39)**	**72.34** **(0.09)**	**71.49** **(0.08)**	**95.04** **(0.03)**	**69.28** **(0.06)**

Table 5: Performance on link prediction with random and hard negative edges.

		Computers		Photo		Co. CS		Co. Physics	
		AUC	AP	AUC	AP	AUC	AP	AUC	AP
Random Neg.	GRACE	0.939	0.939	0.962	0.960	0.970	0.970	OOM	OOM
	GCA	0.954	0.954	0.965	0.960	**0.971**	**0.970**	OOM	OOM
	CCA-SSG	0.961	0.959	0.973	0.970	0.949	0.950	0.943	0.936
	BGRL	0.964	0.961	0.978	0.976	0.952	0.948	0.952	0.947
	RGRL	**0.974**	**0.972**	**0.983**	**0.981**	0.967	0.968	**0.964**	**0.963**
Hard Neg.	GRACE	0.933	0.933	0.939	0.929	0.870	0.868	OOM	OOM
	GCA	0.938	0.929	0.948	0.939	0.874	0.869	OOM	OOM
	CCA-SSG	0.954	0.952	0.947	0.943	0.847	0.835	0.871	0.856
	BGRL	0.959	0.956	0.959	0.956	0.845	0.832	0.903	0.892
	RGRL	**0.969**	**0.968**	**0.967**	**0.964**	**0.878**	**0.881**	**0.923**	**0.919**

Evaluation on link prediction. Table 5 shows the link prediction performance on random and hard negative edges. We have the following observations: **1)** RGRL discovers the true existing relationships between the nodes better than all other baseline methods. This implies that learning augmentation-invariant relationship among nodes is beneficial not only for node classification but also for link prediction. **2)** The improvements of RGRL on hard negative edges, which is a more practical task, is more significant than that on random negative edges implying that RGRL detects more fine-grained relational information compared with other baselines. **3)** In the case of Computers and Photo datasets, non-contrastive methods (i.e., BGRL, CCA-SSG) outperform contrastive methods (i.e., GRACE, GCA), while this is not the case for Coauthor CS and Physics datasets. On the other hand, RGRL outperforms all other baseline methods regardless of the datasets, since RGRL relaxes the strict constraints of both contrastive/non-contrastive methods thereby achieving the best of both worlds as discussed in Sec. 5.

Evaluation on multiplex network. Table 6 shows the node classification performance on multiplex networks. RGRL outperforms previous state-of-the-art methods for multiplex networks showing the generality of our relational learning framework. To demonstrate the generality of RGRL, we apply BGRL to multiplex networks closely following the extension of RGRL. We observe that RGRL outperforms all other baseline methods whereas BGRL shows competitive performance with baseline methods. We argue that this is mainly due to the strict constraint of BGRL in that BGRL learns layer-invariant features whereas RGRL learns layer-invariant relationships as discussed in Sec. 5. Since RGRL allows the node representations to vary as long as the core relationship among the multiple layers is kept, we argue that RGRL can learn from more diverse relationship inherent in multiplex network.

Table 6: Performance on multiplex network.

Dataset	IMDB		DBLP	
Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1
HAN	0.599	0.607	0.716	0.708
DMGI	0.648	0.648	0.771	0.766
DMGI$_{\text{attn}}$	0.602	0.606	0.778	0.770
HDMI	0.650	**0.658**	0.820	0.811
BGRL	0.631	0.634	0.819	0.807
RGRL	**0.653**	**0.658**	**0.830**	**0.818**

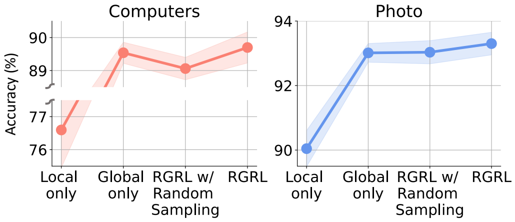


Figure 6: Ablation studies.

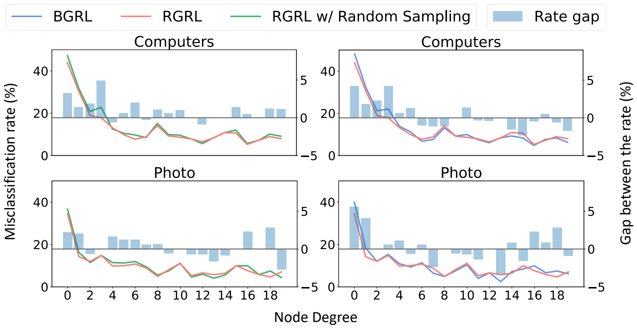


Figure 7: Comparisons of misclassification rate per node degree. (right) RGRL vs. RGRL with random sampling (left) RGRL vs. BGRL. Rate gap indicates how well RGRL performs compared with the baseline.

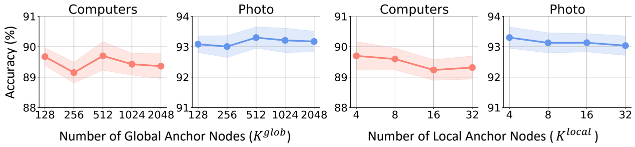


Figure 8: Sensitivity analysis on number of global anchor nodes (Left) and local anchor nodes (Right).

6.3 Ablation Studies

To verify the benefit of each component of RGRL, we conduct ablation studies on two datasets, i.e., Amazon Computers and Amazon Photo in Fig. 6. We have the following observations: **1)** Considering the global similarity is more beneficial than considering the local similarity. We argue that this is because the global similarity inherently reflects the similarity with both local and non-local nodes thanks to the anchor sampling process, whereas local similarity only reflects the local information. **2)** However, considering the local similarity in addition to the global similarity, i.e., RGRL, shows the best performance. This is because the local similarity module captures the fine-grained relationship among structurally close nodes. **3)** RGRL outperforms "*RGRL w/ Random Sampling*. This implies that RGRL successfully alleviates the degree-bias issue by sampling anchor nodes from the inverse degree-weighted distribution. To verify this, we plot the misclassification rate of RGRL per node degree, and compare it with "*RGRL w/ Random Sampling* in Fig. 7 (left). We observe that RGRL outperforms "*RGRL w/ Random Sampling* on low-degree nodes while competitive on other nodes, demonstrating that sampling anchor nodes from the inverse degree-weighted distribution helps alleviate degree-bias issue. **4)** Moreover, we also compare the misclassification rate of RGRL per node degree with BGRL in Fig. 7 (right). We observe that RGRL greatly outperforms BGRL especially on low-degree nodes, indicating the superiority of the relation-preserving loss of RGRL over the self-preserving loss of BGRL for alleviating the degree-bias issue of GNNs. As most real-world graphs are long-tailed, i.e., a majority of nodes have low degree, we argue that RGRL is practical for use in reality.

6.4 Sensitivity Analysis

On the number of sampled anchor nodes. Fig. 8 shows the sensitivity analysis on the number of local and global anchor nodes, i.e., K^{local} and K^{glob}. We have the following observations: **1)** Regarding the number of global anchor nodes, i.e., K^{glob}, we observe that the performance of node classification is relatively stable over various K^{glob}s. In other words, sampling more global anchor nodes does not contribute much to the model performance. We argue that since different global anchor nodes are sampled in every training epoch, the computed similarity distributions would be diverse, which in turn facilitates the model to learn from diverse supervisory signals even with a small number of sampled nodes. Thanks to its robustness to the number of global anchor nodes, RGRL is practical for applying on large-scale graphs common in real-world, such as *ogbn-arXiv* in Table 4. **2)** Regarding the number of local anchor nodes, i.e., K^{local}, we observe a slight performance degradation as we increase K^{local}. This is expected since more false positives are introduced to $\mathrm{N}_i^{\text{local}}$ as K^{local} gets larger as shown in Fig. 4.

On the temperature hyperparameters. Fig. 9 shows the sensitivity analysis on the temperature hyperparameters τ_ξ^{glob} and τ_ξ^{local} of the target encoder. Note that temperature hyperparameters (i.e., $\tau_\theta^{\text{glob}}$ and $\tau_\theta^{\text{local}}$) of the online network are fixed to 0.1. We have the following observations: **1)** Regarding the global temperature hyperparameters, we observe that the best τ_ξ^{glob} is 0.01 in both datasets, which is smaller than $\tau_\theta^{\text{glob}}$ that is fixed to 0.1. Note that the target distribution gets sharper as τ_ξ^{glob} gets smaller. This aligns with the argument in [39] that target distribution should be sharpened to provide a stronger supervisory signal for the model training. Thus, with τ_ξ^{glob} smaller than $\tau_\theta^{\text{glob}}$, RGRL can learn more discriminative node representations. **2)** On the other hand, regarding the local temperature hyperparameters, we observe that a high value of τ_ξ^{local} is more beneficial for the model performance, i.e., $\tau_\xi^{\text{local}} = 1.0$ shows the best performance in both datasets. Since target distribution p_ξ^{local} gets softened as τ_ξ^{local} gets larger, the model is trained to be less discriminative among the sampled local anchor nodes $\mathrm{N}_i^{\text{local}}$. That is, the model is trained to learn a set of local anchor nodes $\mathrm{N}_i^{\text{local}}$, which are structurally close and share the same semantic, to be close in representation space. As shown in Fig. 10, given a

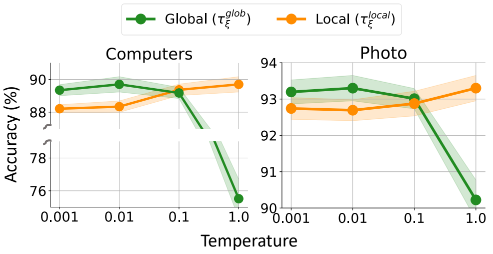


Figure 9: Sensitivity analysis on temperature.

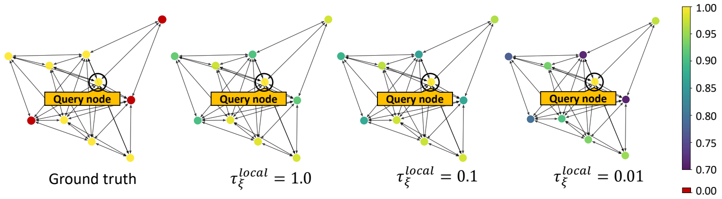


Figure 10: How τ_ξ^{local} reflects the structural information around the query node. Color indicates the cosine similarity between the query node and its neighboring nodes. For Ground truth, the similarity is 1 if the neighboring node shares the same label with query node, and 0 in otherwise.

query node, the cosine similarities of its structurally close nodes are directly correlated with the choice of τ_ξ^{local}; the similarity with the structurally close nodes increases as τ_ξ^{local} gets larger. This implies that RGRL can adaptively choose an appropriate τ_ξ^{local} regarding how much fine-grained relationship should be preserved in the representation space.

6.5 Qualitative Analysis

We conduct a qualitative analysis to further demonstrate the effectiveness of RGRL in capturing the relational information. Our goal is to show the superiority of RGRL over BGRL at 1) discovering core relationships from the given graph, and 2) discovering meaningful knowledge that is not revealed in the given graph. For our analysis, we take “Jiawei Han” and “Christos Faloutsos” as the query authors, both of whom are renowned professors in the field of data mining.

Case 1) In Table 7a, we present a case study that shows which author is the most similar based on the cosine similarity of the learned node representations. In both cases, we observe that the top-1 similar authors discovered by RGRL indeed have more co-authored papers[10] with the query authors compared with BGRL. Surprisingly, the top-1 similar authors discovered by RGRL happen to be former Ph.D. students of the query authors. Considering that the advisor-advisee relationship is one of the core relationships in the academia network that should be preserved no matter how the graph is perturbed, we argue that the relation-preserving framework of RGRL is effective. **Case 2)** Table 7b shows the top-1 similar authors among the authors who do not have any co-authored papers with the query authors. In the case of “Jiawei Han”, the top-1 similar author discovered by BGRL is “Zhou Aoying” whose main research keyword[11] is “Query Processing, whereas that discovered by RGRL is “Ee-Peng Lim” whose main research keyword is “Data & Text Mining”, which is more relevant to the main research topic of “Jiawei Han.” In the case of “Christos Faloutsos”, BGRL and RGRL discovered “Michael J. Pazzani” and “David Jensen” as the top-1 similar authors, respectively, both of whose main research topic is “Machine Learning.” However, it was interesting to see that “David Jensen”, who was discovered by RGRL, actually co-authored two papers [15, 26] with “Christos Faloutsos”, even though this information was missing in the Co.CS dataset. On the other hand, there was no co-authorship between “Christos Faloutsos” and “Michael J. Pazzani”, who is the top-1 similar author discovered by BGRL. Thus, we argue that RGRL discovers meaningful knowledge that is not revealed explicitly in the given graph.

Table 7: Case studies on Coauthor CS dataset.

(a) Case 1: Which author is the most similar?

Query Author	Model	Top-1 Similar Author	# Co-authored Papers	Student?
Jiawei Han	BGRL	Ke Wang	14	✗
	RGRL	Xifeng Yan	87	✓
Christos Faloutsos	BGRL	Tina Eliassi-Rad	27	✗
	RGRL	Hanghang Tong	47	✓

(b) Case 2: Which author will be connected in the future?

Query Author	Model	Top-1 Similar Author	# Co-authored Papers	Research Keywords
Jiawei Han	BGRL	Zhou Aoying	0	Query Processing
	RGRL	Ee-Peng Lim	0	Data & Text Mining
Christos Faloutsos	BGRL	Michael J. Pazzani	0	Machine Learning
	RGRL	David Jensen	2	Machine Learning

7 CONCLUSION

In this paper, we propose a self-supervised learning framework for graphs, named RGRL, which learns node representations such that the relationship among nodes is invariant to augmentations, i.e., augmentation-invariant relationship. By doing so, RGRL allows the node representations to vary as long as the relationship among the nodes is preserved. RGRL learns diverse global relational information among the nodes considering the overall graph structure, while learning fine-grained relationship among structurally close nodes. We also present in-depth discussions on how RGRL achieves the best of both worlds of contrastive/non-contrastive methods by relaxing strict constraints of previous methods with relational information of graph-structured data. Extensive experiments demonstrate that RGRL consistently outperforms existing state-of-the-art methods. Moreover, RGRL 1) demonstrates robustness to less informative or noisy features, and 2) improves performance on low-degree nodes, verifying its practicality in real-world applications.

Acknowledgement. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00157, 100%).

[10] Since Co.CS dataset only provides the co-author relationship among authors, but not how many papers they co-authored, we tried our best to count the number of co-authored papers based on “Google Scholar” along with other resources such as authors' websites.

[11] Authors' research keywords are obtained from “Google Scholar” except for “Zhou Aoying”, whose information is obtained from “ACM Digital Library” due to the lack of information in “Google Scholar.”

REFERENCES

[1] Aleksandar Bojchevski and Stephan Günnemann. 2017. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[4] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased Contrastive Learning. In *NeurIPS*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *In NeurIPS*.

[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

[8] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

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

[10] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687* (2020).

[12] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. HDMI: High-order deep multiplex infomax. In *WWW*.

[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[14] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. *In NeurIPS*.

[15] Ravi Kumar, Alexander Tuzhilin, Christos Faloutsos, David Jensen, Gueorgi Kossinets, Jure Leskovec, and Andrew Tomkins. 2008. Social networks: looking ahead. In *KDD*.

[16] Junseok Lee, Yunhak Oh, Yeonjun In, Namkyeong Lee, Dongmin Hyun, and Chanyoung Park. 2022. GraFN: Semi-Supervised Node Classification on Graph with Few Labels via Non-Parametric Distribution Assignment. *arXiv preprint arXiv:2204.01303* (2022).

[17] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2022. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7372–7380.

[18] Shuai Lin, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric Xing, and Xiaodan Liang. 2021. Prototypical Graph Contrastive Learning. *arXiv preprint arXiv:2106.09645* (2021).

[19] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

[20] Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901* (2020).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report.

[23] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised attributed multiplex network embedding. In *AAAI*.

[24] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*.

[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.

[26] Ted E Senator, Henry G Goldberg, Alex Memory, William T Young, Brad Rees, Robert Pierce, Daniel Huang, Matthew Reardon, David A Bader, Edmond Chow, et al. 2013. Detecting insider threats in a real corporate database of computer usage activity. In *KDD*.

[27] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *TKDE* (2016).

[28] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*.

[29] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *CIKM*.

[30] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. *In ICLR*.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. (2017).

[32] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. In *ICLR*.

[33] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *CVPR*.

[34] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*.

[35] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*.

[36] Jun Xia, Lirong Wu, Jintao Chen, Ge Wang, and Stan Z Li. 2021. Debiased Graph Contrastive Learning. *arXiv preprint arXiv:2110.02027* (2021).

[37] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. 2021. From canonical correlation analysis to self-supervised graph neural networks. *NeurIPS*.

[38] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *NeurIPS* (2018).

[39] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. 2021. ReSSL: Relational Self-Supervised Learning with Weak Augmentation. *NeurIPS* (2021).

[40] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).

[41] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*.

[42] Marinka Zitnik and Jure Leskovec. 2017. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* (2017).