Robust Unsupervised Graph Representation Learning via Mutual Information Maximization

Jihong Wang, Minnan Luo*, Jundong Li, Ziqi Liu, Jun Zhou and Qinghua Zheng

Abstract—Recent studies have shown that GNNs are vulnerable to adversarial attack. Thus, many approaches are proposed to improve the robustness of GNNs against adversarial attacks. Nevertheless, most of these methods measure the model robustness based on label information and thus become infeasible when labels information is not available. Therefore, this paper focuses on robust unsupervised graph representation learning. In particular, to quantify the robustness of GNNs without label information, we propose a robustness measure, named graph representation robustness (GRR), to evaluate the mutual information between adversarially perturbed node representations and the original graph. There are mainly two challenges to estimate GRR: 1) mutual information estimation upon adversarially attacked graphs; 2) high complexity of adversarial attack to perturb node features and graph structure jointly in the training procedure. To tackle these problems, we further propose an effective mutual information estimator with subgraph-level summary and an efficient adversarial training strategy with only feature perturbations. Moreover, we theoretically establish a connection between our proposed GRR measure and the robustness of downstream classifiers, which reveals that GRR can provide a lower bound to the adversarial risk of downstream classifiers. Extensive experiments over several benchmarks demonstrate the effectiveness and superiority of our proposed method.

Index Terms—Adversarial Attacks, Robustness, Unsupervised Graph Representation Learning, Mutual Information

1 INTRODUCTION

Graphs are often used to model interactions between real-world objects and widely appear in a variety of high-impact domains, such as chemistry, social media and bioinformatics [1], [2], [3], [4], [5], to name a few. Owing to the rapid development of deep learning, graph neural networks (GNNs) [6], [7], [8], [9] have made tremendous progress in canonical graph analytical tasks (e.g., node classification, link prediction) in recent years. Note that GNNs often rely on task-specific labels (e.g., labels associated with nodes for the task of node classification) to achieve a decent learning performance. However, such a requirement limits the broad adoptions of GNNs in real-world scenarios as the cost of obtaining label information could be very high. This limitation motivates a surge of research interests in unsupervised graph representation learning [10], [11], [12], [13], [14], which aims to embed the nodes or graphs to a low-dimensional space and extract the most meaningful information for downstream tasks without any label information. Typically, these unsupervised methods utilize GNNs as the encoder function and optimize the encoder with gradient-based algorithms.

Many recent studies [15], [16], [17] have shown that GNNs are vulnerable to adversarial attacks, i.e., even slight perturbations on graph structure or node features may mislead the models to make incorrect predictions. This vulnerability may lead to severe consequences in specific applications. For example, GNNs have been widely used in the recommender systems of many e-commerce platforms [18]. In these platforms, malicious attackers may employ some vicious accounts to interact with the faddish commodity and unpopular commodity synergistically. Thus, they might mislead the GNN-based recommender systems and increase the exposure of unpopular commodities with low quality [21]. Such malicious behaviors may jeopardize user experience and impair recommendation systems.

To address the security and privacy problems caused by the vulnerability of GNNs, recent studies [17], [22], [23] propose several defense methods against graph adversarial attacks, and these efforts are known as robust graph representation learning. However, a vast majority of these studies focus on robust semi-supervised graph representation learning (in which label information must be available) and overlook the vulnerability of unsupervised graph representation learning. Different from semi-supervised methods which rely on label information, robust unsupervised representation learning methods can pretrain the node/graph representations against adversarial attacks without any manually labelled information, i.e., the robustness of learned representations is ensured regardless of any downstream learning tasks.

In this paper, we focus on robust unsupervised graph representation learning. Specifically, given an input graph, we aim to learn a robust encoder that maps nodes to low-dimensional representations, which stay robust even when certain edges as well as node features are perturbed adversarially. To ensure the robustness of the learned node...
representations, there is a fundamental challenge that needs to be addressed: How to quantify the robustness of node representations without any supervised information? For this issue, we propose a measure named graph representation robustness (GRR) to evaluate the robustness of node representations on the basis of mutual information. Specifically, our proposed GRR is defined as the mutual information between the original graph and the node representations extracted from the adversarially perturbed graph. By maximizing the proposed measure of GRR, the perturbed node representations can dwindle the vicious influence of adversarial perturbations and extract more relevant information with the original graph. Furthermore, we theoretically analyze the connection between our GRR measure and downstream tasks (e.g., classification), which show that the proposed GRR provides a lower bound for the adversarial risk \[24\] of downstream classifiers regardless of their specific form.

The challenges of optimizing the proposed GRR measure for robust unsupervised graph representation learning are two-fold. The first challenge centers around mutual information estimation on adversarially perturbed graphs. Although some methods \[10\], \[11\] are proposed to estimate mutual information on graphs, they all assume that the graphs are clean without adversarial attack. Considering that the adversarial perturbations in attacked graphs can increase the difficulty for mutual information estimation, in this paper, we propose a novel mutual information estimator named Subgraph Mutual Information (SMI) upon the adversarially perturbed graph. Our SMI estimates the mutual information with a subgraph-level summary which can provide fine-grained information to help representations eliminate the influence of adversarial perturbations. Experimental results show that our proposed SMI works well on adversarially perturbed graphs. Secondly, perturbing the structure of a graph is a combinatorial problem and thus leads to expensive computational costs. Existing graph adversarial attack methods \[16\], \[17\] try to alleviate this problem by relaxing the discrete adjacency matrix to be continuous and find sub-optimal solutions via gradient-based optimization algorithms. However, it is still computationally expensive to perturb structure in the training procedure. In this regard, we propose an efficient adversarial training strategy with only feature perturbations on the intuition that feature perturbations and structure perturbations behave similarly in GNN neural networks to some extent. Furthermore, a theoretical proof on the simplified graph neural network (SGC) \[25\], and extensive experimental results verify this intuition.

Briefly, the method we propose to maximize GRR with SMI and feature-perturbation-only training strategy is named SMI-GRR, and an overview of our method is provided in Figure 1. All in all, our main contributions of this paper are summarized as follows:

- We propose a novel measure named graph representation robustness (GRR) to evaluate the robustness of node representations without any supervision signals. We also theoretically analyze the connection between our GRR measure and the adversarial risk of downstream tasks (e.g., classification).
- We propose an effective mutual information estimator with a fine-grained subgraph-level summary upon adversarially perturbed graphs, as well as an efficient feature-perturbation-only training strategy for the optimization of GRR.
- Extensive experimental results over five benchmarks demonstrate that our method is capable of learning more robust node representations against adversarial attacks. We also experimentally prove the effectiveness of the proposed mutual information estimator as well as the efficiency of our feature-perturbation-only training strategy.

2 RELATED WORK

In this section, we briefly review related works on two fields closely relevant to our study, i.e., unsupervised graph representation learning and robust graph representation learning.

2.1 Unsupervised Graph Representation Learning

Unsupervised graph representation learning aims to learn low-dimensional representations for nodes in a graph. Traditional methods are usually based on random walk \[26\], \[27\] or matrix factorization techniques \[28\], \[29\]. With the rapid development of graph neural networks (GNNs), numerous unsupervised graph representation learning algorithms \[7\], \[10\], \[30\], \[31\] based on GNNs are proposed.
These methods try to learn representations in an autoencoder manner, i.e., the graph is embedded to low-dimensional node representations by a GNN-based encoder, and then the node representations are optimized by the reconstruction error. For example, Pan et al. [22] proposed to train a GNN-based auto-encoder in an adversarial style to force the representations to follow the Gaussian distribution; GraphSage [7] learns an unsupervised GNN encoder through a random-walk based objective in an inductive way. However, these auto-encoder-based methods overemphasize proximity information [10] and suffer from unstructured predictions [33].

To overcome the shortcomings mentioned above, some works [10, 34, 35, 39, 37] try to learn representations by contrastive learning instead of directly optimizing the reconstruction error. Theoretically, these contrastive learning methods [10, 36] maximize the mutual information instead of overemphasizing proximity information. For example, Zhu et al. [36] proposed to learn node representations with adaptive augmentation that incorporates various priors for topological and semantic aspects of the graph in a contrastive manner. Velickovic et al. [10] proposed to learn node representations through contrasting node and graph encodings and achieve decent performance on several benchmarks for node classification. Despite their effectiveness, existing unsupervised graph representation learning methods mainly focus on learning effective node representations, while the robustness of the learned representations against adversarial attacks is often overlooked.

2.2 Robust Graph Representation Learning

As recent studies [15, 16, 38, 39] reveal that existing graph neural networks are vulnerable to adversarial attacks, robust graph representation learning attracts a surge of interest in recent years. Some defense methods [17, 22, 40, 41, 42, 43] are proposed to eliminate the vicious influence of adversarial examples. According to the used defense strategy, existing defense methods can be divided into three categories, including model-based [17, 22, 40, 41], training-based [17, 40, 41], and preprocessing-based [42, 43] methods. For model-based methods, Ming et al. [22] designed an alternative operator based on graph powering to replace the classical Laplacian in GNN models. They demonstrate that the combination of this operator with vanilla GCN can help defend against evasion attacks. Zhang et al. [23] adopted neighbor importance estimation and layer-wise graph memory components to increase the robustness against various attacks. For training-based methods, Xu et al. [17] proposed a topology attack method based on projected gradient descent, and the attack method is used to improve the robustness of GNNs by adversarial training. Wang et al. [40] leveraged adversarial contrastive learning to improve the robustness of GNN models. The proposed method applies conditional GAN to utilize graph-level auxiliary information. Jin et al. [41] presented Pro-GNN, which jointly learns clean graph structure and trains robust GNN models together. For preprocessing-based methods, Jin et al. [42] proposed to compute the Jaccard Similarity to remove suspicious edges between suspicious nodes. Entezari et al. [43] found that graph adversarial attack tends to generate graphs with high-rank adjacency matrices. Thus, the researchers propose to reduce the effect of attacks by computing the low-rank approximation of the graphs before training GNN models.

Note that all of the previous defense methods against graph adversarial attacks focus on semi-supervised learning, which may be impractical in some situations when labels are rare and expensive. In this paper, we focus on robust graph representation learning without any label information. The most similar work to ours is the work proposed in [44]. The authors propose a robustness measure named graph representation vulnerability (GRV), which does not rely on any supervision signals and achieves competitive learning performance on the adversarially perturbed graph. However, different from their work, our proposed measure theoretically provides a tighter connection to downstream tasks (e.g., classification) (proved in Sec.5). Moreover, the work [44] directly conducts the graph PGD attack [17] on the graph structure, which leads to high complexity, and the adopted mutual information estimator DGI [10] is less effective in adversarial training compared with our proposed method.

3 NOTATION AND PRELIMINARY

In this paper, we use upper-case letters (e.g., X and Y) to denote random variables, the corresponding calligraphic letters (e.g., \(\mathcal{X}\) and \(\mathcal{Y}\)) to denote their support, and the corresponding lower-case (e.g., \(x\) and \(y\)) to denote the realizations of the random variables. \(\mu_X\) and \(\mu_Y\) denote the probability distributions of the corresponding random variables (i.e., \(X\) and \(Y\)). We denote \(\mathbb{P}(X)\) the probability measure on \(X\). Let \((X, \Delta)\) be a metric space, where \(\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}\) is a distance metric. \(B_\Delta(x, \epsilon) = \{x' \in \mathcal{X} : \Delta(x', x) \leq \epsilon\}\) is the ball around \(x\) with radius \(\epsilon\). Moreover, the bold upper-case letters are used to denote matrices (e.g., \(A\) and \(X\)) and bold lower-case letters denote vectors (e.g., \(a\) and \(x\)).

In the framework of unsupervised graph representation learning, an attributed graph is denoted by \(G = (\mathcal{V}, \mathcal{E}, \mathcal{X})\), where \(\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}\) is the node set; \(\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\} \subseteq \mathcal{V} \times \mathcal{V}\) refers to the edge set; Set \(\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{V}|}\}\) collects the feature vectors of all nodes. As we focus on node-level representation learning, we denote the subgraph around a node as a random variable \(S\) with its support denoted by \(\mathcal{S}\). A realization of \(S\) can be formulated as \(s_i = (A_i, X_i)\), where \(A_i, X_i\) denote the adjacency matrix and feature matrix corresponding to a specific node \(i\) and its neighbors. In such a way, the whole graph \(G\) can be seen as a set of the subgraph samples, i.e., \(G = \{s_1, s_2, \ldots, s_n\}\). The goal of node-level unsupervised representation learning is to learn an encoder that maps each subgraph to a representation vector, formally: \(e : \mathcal{S} \to Z\) where \(Z\) is the support of node representations \(Z\). We further define \(f : Z \to \mathcal{Y}\) as a downstream learning task (e.g., classification) that maps a representation \(z \in Z\) to a class label, i.e., \(f(z) \in \mathcal{Y}\). In this sense, \(f \circ e\) denotes the composition of \(f\) and \(e\) such that \((f \circ e)(S) = f(e(S))\).

For a better representation, we recall some definitions our work builds upon as follows.

Definition. (Mutual information [45]) Let \((X, Z)\) be a pair of random variables with values over the space \(\mathcal{X} \times \mathcal{Z}\). The mutual
information of \((X, Z)\) that measures mutual dependence is defined as

\[
I(X, Z) = \int \int_{X} p_{XZ}(x, z) \log \left( \frac{p_{XZ}(x, z)}{p_X(x)p_Z(z)} \right) dx dz, \tag{1}
\]

where \(p_{XZ}\) is the joint probability density function of random variables \((X, Z)\); \(p_X\) and \(p_Z\) are the marginal probability density functions of \(X\) and \(Z\), respectively.

**Definition.** \((p\text{-th Wasserstein distance} \ [46])\) Let \((X, \Delta)\) be a metric space with a bounded support. Given two probability measures \(\mu, \mu'\) on \((X, \Delta)\), the \(p\)-th Wasserstein distance for any \(p \geq 1\) is defined as

\[
W_p(\mu, \mu') = \left( \inf_{\pi \in \Pi(\mu, \mu')} \int_{X \times X} \Delta(x, x) \rho \left( \mu(\cdot) \right) dx \right)^{1/p}, \tag{2}
\]

where \(\Pi(\mu, \mu')\) is the collection of all probability measures on \(X \times X\) with marginals \(\mu\) and \(\mu'\). Specifically, the \(\infty\)-Wasserstein distance is defined as the limit of \(p\)-th Wasserstein distance, i.e.,

\[
W_\infty(\mu, \mu') = \lim_{p \to \infty} W_p(\mu, \mu').
\]

According to the definition of \(\infty\)-Wasserstein distance above, the \(\infty\)-Wasserstein ball with respect to distribution \(\mu\) is defined as

\[
B_{W_\infty}(\mu, \epsilon) = \{ \mu' \in \mathcal{P}(X) : W_\infty(\mu', \mu) \leq \epsilon \}, \tag{3}
\]

where \(\epsilon > 0\) is a pre-defined budget.

### 4 METHODOLOGY

In this section, we first introduce a novel unsupervised robustness measure based on mutual information, and then to optimize the robustness of unsupervised representation learning effectively and efficiently, we introduce our mutual information estimator and training strategy.

#### 4.1 Graph Representation Robustness

Robust unsupervised graph representation learning aims to learn robust node representations of a graph against adversarial attacks without supervision signals. Unlike semi-supervised learning on graphs, quantifying the robustness of node representations without label information is often an intractable problem. In this paper, we propose an unsupervised measure to quantify the robustness of node representations against adversarial perturbations, namely graph representation robustness (GRR).

**Definition.** Given an encoder \(e : S \to Z\) that maps nodes of graph \(G\) with the corresponding subgraphs into low-dimensional representations, graph representation robustness (GRR) is defined as

\[
\text{GRR}_e(e) = \inf_{\mu_S, \epsilon \in B_{W_\infty}(\mu_S, \epsilon)} I(S; e(S')), \tag{4}
\]

where \(S\) is a subgraph that centers around a node, and \(S'\) is the corresponding perturbed subgraph, \(\mu_S\) and \(\mu_{S'}\) denotes the probability distributions of \(S\) and \(S'\), respectively.

Note that GRR measures the lowest mutual information between the original clean graph and the representations learned from the adversarially perturbed graph. Higher GRR indicates that the adversarially perturbed representations are more relative with the original clean graph, and thus contain more intrinsic information. However, directly optimizing GRR for better representation might be impractical, as there is usually a trade-off between accuracy and robustness \([47]\). Instead, we formalize the robust unsupervised representation learning with GRR as the following optimization problem

\[
\max_{e} \mathcal{L} = \alpha \cdot I(S; e(S)) + (1 - \alpha) \cdot \text{GRR}_e(e), \tag{5}
\]

where \(\alpha > 0\) is a trade-off hyper-parameter. By optimizing \(I(S; e(S))\) and GRR jointly, we can ensure the encoder learns the representations with both robustness and effectiveness. The challenge of addressing optimization problem Equation (5) is two-fold: (i) Mutual information estimation on adversarially perturbed graphs. (ii) High complexity to generate adversarial perturbations on graphs, i.e., the infimum over \(S'\) in Equation (4). To these issues, we propose an effective mutual information estimator upon adversarially perturbed graphs and an efficient feature-perturbation-only training strategy.

#### 4.2 Mutual Information Estimation

The estimation of the mutual information for high-dimensional data is intractable as it involves integral; moreover, the prior distribution of the data is usually unreachable, especially for graph which is in non-Euclidean space. Recently some methods \([10, 11]\) are proposed to estimate mutual information on graphs with the assumption that the graphs are clean without adversarial perturbations, and their performance on adversarially perturbed graphs is unstudied. Considering that the adversarial perturbations in attacked graphs can increase the difficulty for mutual information estimation, in this section, we propose a novel mutual information estimator named Subgraph Mutual Information (SMI) upon the adversarially perturbed graph. Our SMI estimates the mutual information with a subgraph-level summary which can provide fine-grained information to help representations eliminate the influence of adversarial perturbations. Specifically, the mutual information \(I(S; e(S'))\), \(I(S; e(S))\) can be estimated based on the Jensen Shannon divergence (JSD) MI estimator \([48]\) respectively

\[
I^{JSD}(S; e(S')) = \text{E}_S[\log D(e(S'), S)],
\]

\[
+ \text{E}_S[\log (1 - D(e(S'), S))], \tag{6}
\]

\[
I^{JSD}(S; e(S)) = \text{E}_S[\log D(e(S), S)],
\]

\[
+ \text{E}_S[\log (1 - D(e(S), S))], \tag{7}
\]

where \(\tilde{S}\) is the negative examples corresponding to \(S\) (usually generated by randomly permuting the nodes in \(S\)). The discriminator \(D\) decides whether the representations and the subgraphs are correlated, which consists of a simplified graph convolution network (SGC) \([25]\) and a bilinear network. The SGC is leveraged to extract the subgraph-level fine-grained summaries from the subgraphs. The bilinear network is adopted to discriminate whether the summary and the node representations are correlated. Specifically, given subgraph \(\tilde{S}\), the discriminator \(D\) can be formalized as

\[
D(e(S), S) = \text{Bilinear}(\text{SGC}(S), e(S)). \tag{8}
\]
Having above mutual information estimator SMI, we reformulate the optimization problem (Equation (5)) as
\[
\max_{e,D} \min_{\mu,\varphi \in \mathcal{B}_{\mu_\infty}(\mu,S,e)} \mathcal{L}_{MI} = \alpha * I^{JS}(S; e(S)) + (1 - \alpha) * I^{JS}(S; e(S')).
\]
(9)
Note that the above optimization problem is bi-level. As we mentioned above, the inner minimization of \( \mathcal{L}_{MI} \) can be regarded as an adversarial attack problem that is computationally expensive as it involves gradients w.r.t. the adjacency matrix. To this issue, we exploit an efficient training strategy subsequently.

4.3 Perturbations Generation

The problem in Equation (9) is a bi-level optimization problem. In the inner minimization problem, we try to get perturbed subgraphs that minimize the mutual information \( I(S, e(S')) \), which is also called the graph adversarial attack problem. Unlike other domains like images, adversarial attack on graphs can be achieved by manipulating graph structure and node features. Compared with node features, the structure of a graph is highly discrete and thus leads to a combinatorial optimization problem. To address this problem, existing works \[15, 17\] usually adopt the gradients of the adjacency matrix to optimize the perturbations. However, these methods can still be computationally expensive for large graphs, making it impractical to perturb structure in the training procedure. Considering that the graph neural networks usually extract information in a message passing way, perturbations on structure and features may behave similarly: perturbations on structure change the path of message propagation while perturbations on features change the message directly. For example, it is evident to achieve the following Theorem 1 for simplified graph convolution neural network (SGC) \[25\].

**Theorem 1.** Given a simplified graph convolution neural network \( g(A; X; \Theta) = \hat{A}^K X \Theta \), where \( \hat{A} = \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} \) is the symmetric normalized adjacency matrix with adjacency matrix with self-loop \( \hat{A} = A + I \) and its degree matrix \( \hat{D} \); \( K \) refers to the number of layers of SGC, and \( \Theta \) denotes the parameter matrix. For any perturbations \( \Delta A \) on the adjacency matrix, there must be corresponding perturbations \( \Delta X \) on features, such that \( g(A + \Delta A; X; \Theta) = g(A; X + \Delta X; \Theta) \).

**Proof.** The theorem can be easily proved by substituting the following equation into Equation (10): \( \Delta X = \left( \hat{A}' \right)^K \left( \hat{A}' \right)^K X - X \). (11)
where \( \hat{A}' \) is the symmetric normalized adjacency matrix of the perturbed adjacency matrix \( \hat{A}' = A + \Delta A; \hat{A}^{-1} \) denotes the pseudo-inverse of \( \hat{A} \). The proof is completed.

Theorem 1 inspires us that perturbations on features may behave equivalently to perturbations on the structure. Thus only considering adversarial attack on features can be a practical way for adversarial training on graph neural networks. Compared with perturbing structure and features jointly, only perturbing features is much more efficient as there is no need to compute the gradients of the adjacency matrix.

To generate perturbations on features, we adopt the projected gradient descent (PGD) method \[49\]. However, the Wasserstein distance ball constraint is still intractable for high-dimensional data in Equation (4). As a result, we consider a subset of \( B_{\mu_\infty}(\mu,S,e) \), denoted by \( B(\mu_S,e) = \{(A_i, X_i) \mid \|X_i - X_i'\|_{2,\infty} < \epsilon, i \in V\} \), (12)
where \( \| \cdot \|_{2,\infty} \) denotes the \( \ell_{2,\infty} \) norm of a matrix, i.e., the maximum \( \ell_2 \) norm of the rows of the matrix; \( V \) refers to the node set of the whole graph; \( A_i, X_i \) are the adjacency matrix and feature matrix of subgraph corresponding to node \( i \), respectively; \( \epsilon \) is used as the adversarial radius in this paper. To further accelerate the perturbing procedure in practice, we conduct PGD attack on the whole graph directly instead of perturbing each subgraph separately. This simplification leads to a suboptimal solution for the adversarial attack problem but reduces the high computational cost on subgraph extraction, which can be expensive when the graph is large.

The whole procedure of our unsupervised robust graph representation learning method is summarized in Algorithm 1 which is named SMI-GRR. The training procedure mainly consists of two steps: adversarial attack step (line 6) to get the perturbed subgraphs and the optimizing step (line 7) to optimize the mutual information estimator with the perturbed subgraphs. Note that calculations on subgraphs can be easily parallelized with matrix operation.

4.4 Complexity Analysis

The main computational cost for our method is the adversarial attack step (line 6 in Algorithm 1). Thus, we theoretically compare the efficiency of two training strategies: training with feature-only perturbations and training with joint perturbations. The main difference is the calculation of the gradients. For training with feature-only perturbations, the adjacency matrix can be stored in a sparse form. Thus the memory complexity is \( O(m + nd) \) where \( m \) denotes the number of edges, \( n \) is the number of nodes, and \( d \) is...
the dimension of node features. For time complexity, as we only need to calculate the gradients of features, it can be $O(n_{\text{iter}} \times nd)$ where $n_{\text{iter}}$ is the number of iterations for PGD attack conducted on features. For training with joint perturbations, the graph PGD attack method [17] is adopted to perturb structure. The adjacency matrix has to be stored in a dense form as the graph PGD attack involves gradients w.r.t. adjacency matrix, and thus the memory complexity can be $O(n^2 + nd)$. For time complexity, all entries in the adjacency matrix should be considered, and the time complexity can be $O(T \times n^2 + n_{\text{iter}} \times nd)$ where $T$ is the number of iterations for the graph PGD attack method. The complexity is summarized in Table 1. Note that $m \ll n^2$, we assert that training with feature-only perturbations is much more efficient than the training strategy with joint perturbations. Moreover, we experimentally prove this point in the experiments section.

5 THEORETICAL ANALYSIS

In this section, we aim to establish a theoretical connection between the unsupervised graph representation robustness GRR and the robustness of the downstream learning task (e.g., classification). To this end, we introduce two lemmas and a definition first.

Lemma 1. (Fano’s Inequality). Let $Y$ be a random variable uniformly distributed over a finite set of outcomes $\mathcal{Y}$. For any estimator $\hat{Y}$ such that $Y \rightarrow X \rightarrow \hat{Y}$ forms a Markov chain, we have

$$\text{Pr}(\hat{Y} \neq Y) \geq 1 - \frac{I(Y; \hat{Y}) + \log 2}{\log |\mathcal{Y}|}. \quad (13)$$

Lemma 2. (Data-Processing Inequality). For any Markov chain $X \rightarrow Y \rightarrow Z$, we have

$$I(X; Y) \geq I(X; Z) \text{ and } I(Y; Z) \geq I(X; Z). \quad (14)$$

Definition. (Adversarial Risk). Let $(X, \Delta)$ be the input metric space and $\mathcal{Y}$ be the set of labels. Let $\mu_{XY}$ be the underlying distribution of the input and label pairs. For any classifier $f : X \rightarrow \mathcal{Y}$, the adversarial risk of $f$ with respect to $\epsilon \geq 0$ is defined as

$$\text{AdvRisk}_\epsilon(f) = \text{Pr}_{(x, y) \sim \mu_{XY}}[\exists x' \in B(x, \epsilon) \text{ s.t. } f(x') \neq y]. \quad (15)$$

Adversarial risk measures the vulnerability of a given classifier to adversarial perturbations. The lower the adversarial risk is, the more robust the classifier $f$ is. We prove a theoretical connection between GRR and the adversarial risk of downstream classifiers. Specifically, GRR provides a lower bound for the adversarial risk regardless of the specific form of downstream classifiers.

Theorem 2. Let $(S, \Delta)$ be the input metric space, $\mathcal{Y}$ be the set of labels and $\mu_{SY}$ be the underlying joint probability distribution. Assume the marginal distribution of labels $\mu_Y$ is a uniform distribution over $\mathcal{Y}$. Consider the feature space $Z$ and the set of downstream classifier $\mathcal{F} = \{f : Z \rightarrow \mathcal{Y}\}$. Given $\epsilon \geq 0$, for any $f : S \rightarrow Z$, the following inequality holds

$$\inf_{f \in \mathcal{F}} \text{AdvRisk}_\epsilon(f \circ o) \geq 1 - \frac{\text{GRR}(\epsilon) + \log 2}{\log |\mathcal{Y}|}. \quad (16)$$

Proof. For any given $\mu_S \in \mathcal{B}_{W_{\infty}}$ and $f \in \mathcal{F}$, we have the following Markov chain $Y \rightarrow S \rightarrow S' \rightarrow f(e(S')) \rightarrow (f \circ o)(S')$, where $S, Y$ are random variables for input and label distributions respectively. By applying Lemma 1 and Lemma 2, we arrive at the following inequality

$$\text{Pr}[f(e(S')) \neq Y] \geq 1 - \frac{I(S; e(S')) + \log 2}{\log |\mathcal{Y}|}. \quad (18)$$

Note that the right side of Equation (20) is irrelevant with $f$, thus the infimum over $f$ can be omitted. By taking the irrelevant item out of the supremum, we can finally derive the conclusion

$$\inf_{f \in \mathcal{F}} \text{AdvRisk}_\epsilon(f \circ o) \geq \inf_{f \in \mathcal{F}} \sup_{\mu_S \in \mathcal{B}_{W_{\infty}}} \left[1 - \frac{I(S; e(S')) + \log 2}{\log |\mathcal{Y}|}\right]. \quad (20)$$

Substituting Equation (18) into Equation (19), we get a lower bound of adversarial risk

$$\inf_{f \in \mathcal{F}} \text{AdvRisk}_\epsilon(f \circ o) \geq 1 - \frac{\text{GRR}(\epsilon) + \log 2}{\log |\mathcal{Y}|}. \quad (21)$$

The proof is completed. \hfill \square

From Theorem 2, we can find that GRR provides a lower bound for the minimum of adversarial risk. In other words, any downstream classifier cannot be robust if the GRR is low. This connection indicates that we can get robust node representations without the knowledge of downstream classifiers and label information by maximizing GRR instead.

Recently, a similar work [44] also proposes a measure named graph representation vulnerability (GRV) to measure the robustness of graph representations, i.e.,

$$\text{GRV}(\epsilon) = I(S; e(S)) - \inf_{\mu_S \in \mathcal{B}(\mu_S, \epsilon)} I(S'; e(S')). \quad (22)$$

GRV tries to maximize the mutual information between perturbed representations and the perturbed graph, i.e., GRV focuses on the robustness of the mutual information estimator instead of the robustness of node representations. Theoretically, compared with GRR, our GRR provides a tighter lower bound for the adversarial risk of downstream classifiers.
Theorem 3. Let \((S, \Delta)\) be the input metric space, \(\mathcal{Y}\) be the set of labels and \(\mu_{SY}\) be the underlying joint probability distribution. Assume the marginal distribution of labels \(\mu_Y\) is a uniform distribution over \(\mathcal{Y}\). Consider the feature space \(Z\) and the set of downstream classifier \(\mathcal{F} = \{f : Z \to \mathcal{Y}\}\). Given \(\epsilon \geq 0\), for any \(e: S \to Z\), we have:

\[
\inf_{f \in \mathcal{F}} \text{AdvRisk}_e(f \circ e) \geq 1 - \frac{\text{GRR}_e(e) + \log 2}{\log |\mathcal{Y}|}.
\]

(23)

where \(S\) is the random variable that follows the marginal distribution of inputs \(\mu_S\).

Proof. The first inequality is from Theorem 2. The remaining task of the proof is to prove the second inequality. Considering Lemma 2 and the Markov chain in Equation (17), we have

\[
I(S; e(S')) \leq I(S', e(S')).
\]

(24)

Taking the infimum on both side of the inequality above, we arrive at

\[
1 - \frac{\inf_{\mu_{SE} \in B_{W_{\infty}}} I(S; e(S')) + \log 2}{\log |\mathcal{Y}|} \geq 1 - \frac{\inf_{\mu_{SE} \in B_{W_{\infty}}} I(S'; e(S')) + \log 2}{\log |\mathcal{Y}|}.
\]

(25)

Note that the above inequality is an expansion of the second inequality in Equation (23). The proof is completed.

Experimental results in the experiments section also confirm the theoretical analysis, which show that our proposed GRR outperforms GRV upon adversarially attacked graphs. Moreover, compared with the work [44], our method not only provides a tighter connection with the adversarial risk of downstream classifiers but also provides a more effective mutual information estimator and a more efficient training strategy, as we have stated above. We will empirically show all these points in the next section.

6 EXPERIMENTS

In this section, we empirically evaluate the robustness of our proposed method. In particular, we aim to answer the following three questions:

- **RQ1** How does our proposed method perform compared to other baseline methods when the graph is adversarially attacked?
- **RQ2** Do our proposed adversarial training strategy and mutual information estimator benefit the performance of our method as expected?
- **RQ3** How do different hyper-parameters affect the performance of our method?

Before presenting our experimental results and observations, we first introduce the experimental setup.

### 6.1 Experimental Setup

#### 6.1.1 Datasets

We conduct experiments on five benchmark datasets, including Cora, Citeseer, Pubmed, Cora ML, and CoauthorCS. The first four datasets are collected from citation networks where the nodes correspond to documents and edges indicate citation relationships. The CoauthorCS dataset is an academic network containing co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge. In this graph, nodes represent authors, and edges indicate co-authorship relationships. The statistical details are shown in Table 2. For Cora, Citeseer, and Pubmed, we follow the standard data split [6]. For Cora ML and CoauthorCS, the nodes are randomly split to train, validation, and test sets with 10%, 10% and 80% of the nodes, separately.

| Dataset     | #Nodes | #Edges | #Features | #Classes | Degree |
|-------------|--------|--------|-----------|----------|--------|
| Cora        | 2,708  | 5,278  | 1,433     | 7        | 4.90   |
| Citeseer    | 3,327  | 4,552  | 3,703     | 6        | 3.74   |
| Cora ML     | 2,995  | 8,158  | 2,879     | 7        | 6.45   |
| Pubmed      | 19,717 | 44,324 | 500       | 3        | 5.50   |
| CoauthorCS  | 18,333 | 81,894 | 6,805     | 15       | 8.93   |

#### 6.1.2 Baselines

As far as we know, there are no existing methods focusing on unsupervised robust representation learning except for [44]. Thus we consider four non-robust graph representation learning methods, two preprocessing-based defense methods against the adversarial attacks, and our proposed method:

- **GAE** [30]. GAE is a widely used unsupervised method for graph data, which learns the representations of nodes with a GCN based auto-encoder.
- **DGI** [10]. DGI learns the node representations by mutual information maximization.
- **MVGRL** [37]. MVGRL is a self-supervised approach for node-level and graph-level representation learning by contrasting multiple structural views of graphs.
- **GCA** [36]. GCA is a graph contrastive representation learning method with adaptive augmentation that incorporates various priors for topological and semantic aspects of the graph.
- **DGI-Jaccard** [42]. DGI-Jaccard is a preprocessing-based method that drops the edges with low Jaccard similarity, and then DGI is applied to learn the representations on the graph preprocessed.
- **DGI-SVD** [43]. DGI-SVD is a preprocessing-based method that denoises the adjacency matrix with a low-rank approximation, and then DGI is applied to learn the representations on the preprocessed graph.
- **SMI-GRV** [44]. Graph representation vulnerability (GRV) is another robustness measure proposed by [44] which is similar to our proposed GRR. We adopt their robustness measure and set the training proce-
dure and mutual information estimator the same as our method for a fair comparison.

6.1.3 Implementation Details

In our experiments, we adopt a two-layer GCN [6] as the encoder. The hidden dimensions of GCN are set as 512 on all datasets. For simplicity, the SGC in the discriminator shares the same parameters with the last layer of GCN. In the preprocessing procedure, the features of nodes are renormalized with $\ell_2$ norm. And in the training procedure, the MI estimator is trained for 300 epochs. The adversarial radius $\epsilon$ is fixed as 0.5 if not explicitly stated. The trade-off parameter $\alpha$ is set to be 0.5 for Cora, Citeseer and Cora_ML, and 0.8 for larger graphs Pubmed and CoauthorCS as higher $\alpha$ help the model converges more quickly. For evaluating the robustness and effectiveness of the node representations, logistic regression is adopted as a downstream classifier. As existing graph adversarial attack methods [15], [17], focus on the semi-supervised node classification task, we combine our unsupervised encoder and downstream classifiers as an end-to-end node classifier for evaluation. Specifically, we first train a logistic regression classifier on the node representations extracted from the original graph. Then the logistic regression classifier and the encoder are combined as an end-to-end node classifier. Next, we conduct adversarial attacks on the end-to-end node classifier and get the attacked node representations by feeding the attacked graph to the encoder. Finally, we retrain 10 new classifiers on the attacked node representations, and the average accuracy is reported. For adversarial attacks on features, we adopt PGD method [9]. Moreover, for attack on structure, we adopt the graph PGD attack [17] for Cora, Citeseer, and Cora_ML, and DICE attack [15] for Pubmed and CoauthorCS. This choice is because the graph PGD attack is a powerful method but with high time complexity and fails to scale to larger graphs, while DICE is a random-based method that is less powerful but more efficient. All nodes in the test set are regarded as target nodes to attack, i.e., we evaluate the robustness of nodes in the test set. Experiments are conducted on a machine with CPU E5-2650 v4 @ 2.20GHz and a 2080 Ti GPU with 12GB RAM.

6.2 Robustness Against Adversarial Attack

In Table 3, we summarize the learning performance of baselines and our method on both benign and attacked graphs. The bold highlighted numbers indicate the best among all methods. For PGD attack on features, the budget is fixed as 0.1, and for the graph PGD attack on structure, we set the budget as $0.5 \times |V_{test}|$ where $|V_{test}|$ denotes the number of test nodes. As the DICE attack conducted on larger graphs is less powerful, the budget is set as 1.0 $\times |V_{test}|$.

From Table 3, we can make the following observations: (i) Just like existing works on semi-supervised methods [15], [16], [17], unsupervised graph representation learning methods are also vulnerable and can be easily misled by adversarial attacks on graphs. For example, the classification accuracy of DGI on Cora decreases from 82.4% to 63.0% after the attacks, and for MVGRL (the SOTA method on the benign graph), its classification performance is significantly impaired from 83.5% to 49.5% after the attacks; (ii) Preprocessing-based defense methods fail in our experimental setting. DGI-Jaccard and DGI-SVD perform similar to DGI after the attacks, and DGI-SVD even performs worse on benign graphs because of the low-rank approximation. This may be because both preprocessing-based methods only consider structure perturbations and are thus powerless for feature perturbations. (iii) Both SMI-GRV and SMI-GRR outperform other methods, which indicates that the unsupervised robustness measures benefit the robustness of unsupervised representation learning against adversarial attacks. Moreover, we can observe that the robustness of SMI-GRV we propose exceeds SMI-GRV, which is consistent with the theoretical analysis above. This observation shows that the tighter lower bound of adversarial risk (i.e., AdvRisk) provided by GRR contributes to the robustness of node representations.

To study the performance of unsupervised graph representation learning methods against adversarial attacks with different budgets, we show the accuracy of different methods with feature perturbation budget ranging from $[0.0, 0.02, 0.04, 0.06, 0.08, 0.1]$ and structure perturbation budget ranging from $[0.0, 0.5, 1.0, 1.5, 2.0, 2.5]$ in Figure 2. From the figure, we can observe that the performance of GAE, MVGRL, and GCA decline greatly. At the same time, DGI and its preprocessing-based variants perform better than the former three unsupervised methods, and our proposed SMI-GRR performs the best. Regarding why DGI outperforms GAE, MVGRL, and GCA, we make the following explanations: (i) Compared with other methods, GAE is a weaker method, and the encoder-decoder architecture makes the learned representations heavily dependent on the graph structure and features. (ii) MVGRL utilizes an MLP projection head after encoding, making the model deeper.
Fig. 2: Accuracy against adversarial attack with different budgets. For (a), (b), (c), the budgets of feature perturbations range from \([0, 0.02, 0.04, 0.06, 0.08, 0.1]\) and structure perturbations are fixed as 0.5. For (d), (e), (f), the budgets of structure perturbations range from \([0, 0.5, 1.0, 1.5, 2.0, 2.5]\) and feature perturbations are fixed as 0.1.

as it involves more layers than other methods without projection head. When the input is attacked, the perturbations will be accumulated in each layer, and thus the deeper model may suffer more disturbance. (iii) GCA augments the input graph adaptively, making the representations depend heavily on the graph structure and features. Moreover, we can observe that both SMI-GRV and SMI-GRR outperform DGI and our proposed SMI-GRR performs the best. This observation corresponds to the above results, which indicates that our conclusion does not depend on specific perturbation budgets.

For an intuitive illustration, we visualize the representations learned from Cora with t-SNE \([53]\) in Table 4. From a qualitative perspective, we can find that the intercluster boundary of DGI gets blurred after adversarial attacks while SMI-GRV and SMI-GRR are more robust. Specifically, for our proposed SMI-GRR method, the intercluster boundary is clearer than other methods. For example, the nodes labeled green, orange, and yellow have wider decision boundaries for SMI-GRR than DGI and SMI-GRV. This observation indicates that our method improves the robustness of representations, and the intercluster boundary can be more stable when the graphs are attacked.

6.3 Impact on Training Strategy and Mutual Information Estimator

In this paper, in addition to a novel robustness measure GRR, we also adopt a more effective mutual information estimator and a more efficient training strategy. We experimentally show that these points benefit the performance of our method.

6.3.1 Effectiveness of Mutual Information Estimator

In Table 5 we compare the performance of different mutual information estimators, i.e., our proposed SMI and the widely used DGI \([10]\). We can observe that SMI outperforms DGI on attacked graphs either with GRR or GRV while two estimators perform similarly on benign graphs. The improvement on performance benefits from the fine-grained summary provided by SMI. Specifically, DGI extracts the graph-level information by a readout function (mean pooling on all node representations), which is coarse-grained and leads to information loss. Our proposed SMI leverage subgraph-level summary (extracted by SGC), which can provide fine-grained information when they are fed into the discriminator with node representations. The fine-grained information can help identify perturbations on node representations more precisely and make the encoder more robust against perturbations. Moreover, we can also observe that GRR outperforms GRV on both mutual information estimators, proving our method’s superiority.

6.3.2 Efficiency of Training Strategy

Instead of directly perturbing the structure and features simultaneously during the training procedure, we think that
both structure and features behave similarly for graph neural networks and only perturbing the features is sufficient for adversarial training. We compare the performance of two different training strategies in Figure 3. The methods with ‘-S’ denote the training strategy with joint perturbations, and methods without ‘-S’ denote the training strategy with feature-only perturbations. We can find that perturbing structure with proper budgets can benefit both GRR and GRV, but the training strategy with only perturbing features is also competitive. For example, SMI-GRR-S performs the best (i.e., accuracy equals 73.1%) when the budget of structure perturbations is set as 2.0 while SMI-GRR gets 71.8%. Moreover, we emphasize that for both training strategies, our method SMI-GRR outperforms SMI-GRV. To study the efficiency of the two training procedures, we report the average execution time and memory cost in 100 epochs in Table 5. As the choice of robustness measure will not affect the complexity of optimization, we only report the execution time of our proposed GRR. We find that only perturbing features in the training procedure is much more efficient than jointly perturbing the structure and features. For example, an epoch will cost 0.35s on Cora if only perturbing features and 2.39s if structure and features are perturbed simultaneously, i.e., it will lead to $6.8 \times$ time cost increase if the structure is perturbed in the training procedure. Furthermore, the joint perturbation training strategy cannot scale to larger graphs like Pubmed and CoauthorCS because of the high memory cost. Considering the slight decline in effectiveness but significant improvement in efficiency, only perturbing features in the training procedure is a more reasonable strategy.
Fig. 3: Accuracy of different training strategies. Methods with ‘S’ denote structure and features are both perturbed in the training procedure and otherwise only feature perturbations are considered.

6.4 Parameter Study

In order to explore the effect of different values of hyperparameters, i.e., the adversarial radius $\epsilon$ and the trade-off parameter $\alpha$, we report the ablation study results on Cora with different values of $\epsilon$ and $\alpha$ in Figure 4 separately.

6.4.1 Influence of Adversarial Radius $\epsilon$

In Figure 4, we report the accuracy of SMI-GRV and SMI-GRR when the adversarial radius $\epsilon$ is changed. We can observe that for both SMI-GRV and SMI-GRR, a lower value of $\epsilon$ (i.e., $\epsilon = 0.1$) leads to lower robustness while the performance on the benign graph is not influenced. This observation accords with the intuition that when $\epsilon$ is low, the adversarial perturbations we consider during the training procedure are insufficient to ensure robustness. In other words, both SMI-GRV and SMI-GRR provide a ‘protective barrier’ with radius $\epsilon$, and it will be easily penetrated if $\epsilon$ is low. Moreover, our SMI-GRR performs better than SMI-GRV as our method provides a tighter connection to the robustness of downstream classifiers.

6.4.2 Sensitivity to Trade-off Parameter $\alpha$

In Equation (5), we achieve a trade-off between robustness and effectiveness by using the parameter $\alpha$. We show the performance of SMI-GRR and SMI-GRV when $\alpha$ ranges in $[0.1, 0.3, 0.5, 0.7, 0.9]$. We can find that when $\alpha$ is small (e.g., $\alpha = 0.1$), i.e., when we focus more on the robustness of node representations during optimization, both the robustness against adversarial attack and effectiveness on the benign graph of SMI-GRR and SMI-GRV are worse. This observation indicates that only optimizing the robustness of node representations is undesirable. Moreover, when $\alpha$ is large (e.g., $\alpha = 0.9$), i.e., when we focus more on the effectiveness of representations during optimization, the robustness against adversarial attack is poor while the effectiveness is slightly better. When an appropriate trade-off is achieved, the robustness against adversarial attacks is improved at a slight cost of effectiveness on the benign graph. Moreover, when the trade-off parameter $\alpha$ is changed, our proposed method SMI-GRR still outperforms SMI-GRV, which is consistent with our theoretical analysis.

7 CONCLUSION

In this paper, we focus on robust unsupervised graph representation learning. To evaluate the robustness of node representations without label information, we propose a novel measure named graph representation robustness (GRR), which estimates the mutual information between the original graph and the adversarially perturbed node representations. Compared with the existing methods, our proposed measure provides a tighter connection with downstream classifiers. Moreover, to optimize the graph representation robustness effectively and efficiently, we propose a mutual information estimator with fine-grained graph summary and a training strategy with only feature perturbation. Experimental results demonstrate the effectiveness and efficiency of our proposed method.
Jihong Wang received the B.Eng. degree from the School of Computer Science and Technology, Xi’an Jiaotong University, China, in 2019. He is currently a Ph.D. student in the School of Computer Science and Technology, Xi’An Jiaotong University, China. His research interests include robust machine learning and its applications, such as social computing and learning algorithms on graphs.

Minnan Luo received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2014. She was a Post-Doctoral Research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. She is currently an Associate Professor in the School of Electronic and Information Engineering at Xi’an Jiaotong University. Her research interests include machine learning and optimization, data mining, image processing, and cross-media retrieval.

Jundong Li received the Ph.D. degree in Computer Science at Arizona State University in 2019, M.Sc. degree in Computer Science at University of Alberta in 2014, and B.Eng. degree in Software Engineering at Zhejiang University in 2012, China. He is an Assistant Professor at the University of Virginia with appointments in Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science. His research interests include data mining, machine learning and graph learning.

Ziqi Liu received the Ph.D. degree from the Department of Computer Science, Xi’an Jiaotong University, China, in 2017. He was a visiting researcher advised by Prof. Alex Smola at Machine Learning Department, Carnegie Mellon University. He currently works at Ant Group. His research interests include probabilistic models, graphical models, nonparametric modeling, large-scale inference algorithms and applications in user modeling, text mining.

Jun Zhou is currently a Senior Staff Engineer at Ant Financial. His research mainly focuses on machine learning and data mining. He has participated in the development of several distributed systems and machine learning platforms in Alibaba and Ant Financial, such as Apsaras (Distributed Operating System), MaxCompute (Big Data Platform), and Kunpeng (Parameter Server). He is a member of IEEE. He has published more than 40 papers in top-tier machine learning and data mining conferences, including VLDB, WWW, SIGIR, NeurIPS, AAAI, IJCAI, and KDD.

Qinghua Zheng received the Ph.D. degree of system engineering in 1997, the M.Sc. degree of computer organization and architecture in 1993, and the B.Eng. degree of computer software in 1990 at Xi’an Jiaotong University, China. He was a Post-Doctoral Researcher with Harvard University, Cambridge, MA, USA, in 2002. He is currently a Professor in the School of Electronic and Information Engineering at Xi’an Jiaotong University. His research interests include computer network security, intelligent E-learning theory and algorithm, multimedia e-learning, and trustworthy software.