Adversarial Graph Contrastive Learning with Information Regularization

Shengyu Feng
shengyu8@illinois.edu
University of Illinois at Urbana-Champaign
USA

Baoyu Jing
baoyuj2@illinois.edu
University of Illinois at Urbana-Champaign
USA

Yada Zhu
yzhu@us.ibm.com
IBM Research
USA

Hanghang Tong
htong@illinois.edu
University of Illinois at Urbana-Champaign
USA

ABSTRACT

Contrastive learning is an effective unsupervised method in graph representation learning. Recently, the data augmentation based contrastive learning method has been extended from images to graphs. However, most prior works are directly adapted from the models designed for images. Unlike the data augmentation on images, the data augmentation on graphs is far less intuitive and much harder to provide high-quality contrastive samples, which are the key to the performance of contrastive learning models. This leaves much space for improvement over the existing graph contrastive learning frameworks. In this work, by introducing an adversarial graph view and an information regularizer, we propose a simple but effective method, *Adversarial Graph Contrastive Learning* (ARIEL), to extract informative contrastive samples within a reasonable constraint. It consistently outperforms the current graph contrastive learning methods in the node classification task over various real-world datasets and further improves the robustness of graph contrastive learning.

CCS CONCEPTS

• **Mathematics of computing** → **Information theory**; **Graph algorithms**; • **Computing methodologies** → **Neural networks**; **Learning latent representations**.

KEYWORDS

graph representation learning, contrastive learning, adversarial training, mutual information

ACM Reference Format:
Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. 2022. Adversarial Graph Contrastive Learning with Information Regularization. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512183

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512183

1 INTRODUCTION

Contrastive learning is a widely used technique in various graph representation learning tasks. In contrastive learning, the model tries to minimize the distances among positive pairs and maximize the distances among negative pairs in the embedding space. The definition of positive and negative pairs is the key component in contrastive learning. Earlier methods like DeepWalk [24] and node2vec [6] define positive and negative pairs based on the co-occurrence of node pairs in the random walks. For knowledge graph embedding, it is a common practice to define positive and negative pairs based on translations [2, 11, 18, 33, 34, 36].

Recently, the breakthroughs of contrastive learning in computer vision have inspired some works to apply the similar ideas from visual representation learning to graph representation learning. To name a few, Deep Graph Infomax (DGI) [32] extends Deep InfoMax [9] and achieves significant improvements over previous random-walk based methods. Graphical Mutual Information (GMI) [23] uses the same framework as DGI but generalizes the concept of mutual information from vector space to graph domain. Contrastive multi-view graph representation learning (referred to as MVGRL in this paper) [7] further improves DGI by introducing graph diffusion into the contrastive learning framework. The more recent works often follow the data augmentation based contrastive learning methods [4, 8], which treat the data augmented samples from the same instance as positive pairs and different instances as negative pairs. Graph Contrastive Coding (GCC) [25] uses random walks with restart [29] to generate two subgraphs for each node as two data augmented samples. Graph Contrastive learning with Adaptive augmentation (GCA) [41] introduces an adaptive data augmentation method which perturbs both the node features and edges according to their importance, and it is trained in a similar way as the famous visual contrastive learning framework SimCLR [4]. Its preliminary work, which uses the uniform random sampling rather than the adaptive sampling, is referred to as GRACE [40] in this paper. Robinson et al. [26] proposes a way to select hard negative samples based on the embedding space distances, and use it to obtain high-quality graph embedding. There are also many works [38, 39] systemically studying the data augmentation on the graphs.

However, unlike the transformations on images, the transformations on graphs are far less intuitive to human beings. The data augmentation on the graph, could be either too similar to or totally

different from the original graph. This in turn leads to a crucial question, that is, *how to generate a new graph that is hard enough for the model to discriminate from the original one, and in the meanwhile also maintains the desired properties?*

Inspired by some recent works [10, 12, 14, 16, 28], we introduce the adversarial training on the graph contrastive learning and propose a new framework called *Adversarial GRaph ContrastIvE Learning* (ARIEL). Through the adversarial attack on both topology and node features, we generate an adversarial sample from the original graph. On the one hand, since the perturbation is under the constraint, the adversarial sample still stays close enough to the original one. On the other hand, the adversarial attack makes sure the adversarial sample is hard to discriminate from the other view by increasing the contrastive loss. On top of that, we propose a new constraint called information regularization which could stabilize the training of ARIEL and prevent the collapsing. We demonstrate that the proposed ARIEL outperforms the existing graph contrastive learning frameworks in the node classification task on both real-world graphs and adversarially attacked graphs.

In summary, we make the following contributions:

- We introduce the adversarial view as a new form of data augmentation in graph contrastive learning.
- We propose the information regularization to stabilize adversarial graph contrastive learning.
- We empirically demonstrate that ARIEL can achieve better performance and higher robustness compared with previous graph contrastive learning methods.

The rest of the paper is organized as follows. Section 2 gives the problem definition of graph representation learning and the preliminaries. Section 3 describes the proposed algorithm. The experimental results are presented in Section 4. After reviewing related work in Section 5, we conclude the paper in Section 6.

2 PROBLEM DEFINITION

In this section, we will introduce all the notations used in this paper and give a formal definition of our problem. Besides, we briefly introduce the preliminaries of our method.

2.1 Graph Representation Learning

For graph representation learning, let $G = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$ be an attributed graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes the feature matrix. Each node v_i has a d-dimensional feature $\mathbf{X}[i,:]$ and all edges are assumed to be unweighted and undirected. We use a binary adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ to represent the nodes and edges information, where $\mathbf{A}[i,j] = 1$ if and only if the node pair $(v_i, v_j) \in \mathcal{E}$. In the following text, we will use $G = \{\mathbf{A}, \mathbf{X}\}$ to represent the graph.

The objective of the graph representation learning is to learn an encoder $f : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d'}$, which maps the nodes in the graph into low-dimensional embeddings. Denote the node embedding matrix $\mathbf{H} = f(\mathbf{A}, \mathbf{X})$, where $\mathbf{H}[i,:] \in \mathbb{R}^{d'}$ is the embedding for node v_i. This representation could be used for downstream tasks like node classification.

2.2 InfoNCE Loss

InfoNCE loss [30] is the predominant work-horse of the contrastive learning loss, which maximizes the lower bound of the mutual information between two random variables. For each positive pair $(\mathbf{x}, \mathbf{x}^+)$ associated with k negative samples of $\mathbf{x}$, denoted as $\{\mathbf{x}_1^-, \mathbf{x}_2^-, \cdots, \mathbf{x}_k^-\}$, InfoNCE loss could be written as

$$L_k = -\log\left(\frac{e^{g(\mathbf{x},\mathbf{x}^+)}}{e^{g(\mathbf{x},\mathbf{x}^+)} + \sum_{i=1}^{k} e^{g(\mathbf{x},\mathbf{x}_i^-)}}\right). \tag{1}$$

Here $g(\cdot)$ is the density ratio with the property that $g(\mathbf{a},\mathbf{b}) \propto \frac{p(\mathbf{a}|\mathbf{b})}{p(\mathbf{a})}$, where $\propto$ stands for *proportional to*. It has been shown by [30] that $-L_k$ actually serves as the lower bound of the mutual information $I(\mathbf{x};\mathbf{x}^+)$ with

$$I(\mathbf{x};\mathbf{x}^+) \geq \log(k) - L_k. \tag{2}$$

2.3 Graph Contrastive Learning

We build the proposed method upon the framework of SimCLR since it is the state of the art contrastive learning method, which is also the basic framework that GCA [41] is built on. Given a graph G, two views of the graph $G_1 = \{\mathbf{A}_1, \mathbf{X}_1\}$ and $G_2 = \{\mathbf{A}_2, \mathbf{X}_2\}$ are first generated. This step can be treated as the data augmentation on the original graph, and various augmentation methods can be used herein. We use random edge dropping and feature masking as GCA does. The node embedding matrix for each graph can be computed as $\mathbf{H}_1 = f(\mathbf{A}_1, \mathbf{X}_1)$ and $\mathbf{H}_2 = f(\mathbf{A}_2, \mathbf{X}_2)$. The corresponding node pairs in two graph views are the positive pairs and all other node pairs are negative. Define $\theta(\mathbf{u},\mathbf{v})$ to be the similarity function between vectors $\mathbf{u}$ and $\mathbf{v}$, in practice, it's usually chosen as the cosine similarity on the projected embedding of each vector, using a two-layer neural network as the projection head. Denote $\mathbf{u}_i = \mathbf{H}_1[i,:]$ and $\mathbf{v}_i = \mathbf{H}_2[i,:]$, the contrastive loss is defined as

$$L_{\text{con}}(G_1, G_2) = \frac{1}{2n}\sum_{i=1}^{n}(l(\mathbf{u}_i,\mathbf{v}_i) + l(\mathbf{v}_i,\mathbf{u}_i)), \tag{3}$$

$$l(\mathbf{u}_i,\mathbf{v}_i) = -\log\frac{e^{\theta(\mathbf{u}_i,\mathbf{v}_i)/\tau}}{e^{\theta(\mathbf{u}_i,\mathbf{v}_i)/\tau} + \sum_{j\neq i} e^{\theta(\mathbf{u}_i,\mathbf{v}_j)/\tau} + \sum_{j\neq i} e^{\theta(\mathbf{u}_i,\mathbf{u}_j)/\tau}}, \tag{4}$$

where τ is a temperature parameter. $l(\mathbf{v}_i,\mathbf{u}_i)$ is symmetrically defined by exchanging the variables in $l(\mathbf{u}_i,\mathbf{v}_i)$. This loss is basically a variant of InfoNCE loss which is symmetrically defined instead.

In principle, our framework could be applied on any graph neural network (GNN) architecture as long as it could be attacked. For simplicity, we employ a two-layer Graph Convolutional Network (GCN) [17] in this work. Define the symmetrically normalized adjacency matrix

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}, \tag{5}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ is the adjacency matrix with self-connections added and $\mathbf{I}_n$ is the identity matrix, $\tilde{\mathbf{D}}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}$ with $\tilde{\mathbf{D}}[i,i] = \sum_j \tilde{\mathbf{A}}[i,j]$. The two-layer GCN is given as

$$f(\mathbf{A},\mathbf{X}) = \sigma(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}), \tag{6}$$

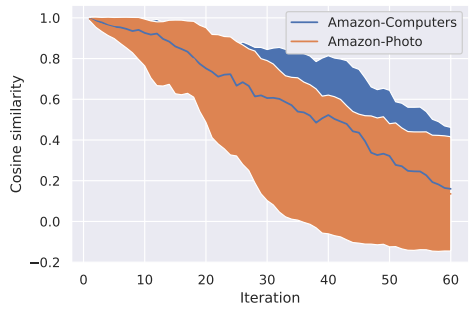

Figure 1: Average cosine similarity between the node embeddings of the original graph and the perturbed graph, results are on datasets Amazon-Computers and Amazon-Photo. The shaded area represents the standard deviation.

where $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are the weights of the first and second layer respectively, $\sigma(\cdot)$ is the activation function.

2.4 Projected Gradient Descent Attack

Projected Gradient Descent (PGD) attack [21] is an iterative attack method which projects the perturbation onto the ball of interest at the end of each iteration. Assuming that the loss $L(\cdot)$ is a function of the input matrix $\mathbf{Z} \in \mathbb{R}^{n\times d}$, at t-th iteration, the perturbation matrix $\mathbf{\Delta} \in \mathbb{R}^{n\times d}$ under an l_∞-norm constraint could be written as

$$\mathbf{\Delta}_t = \Pi_{\|\mathbf{\Delta}\|_\infty \le \delta}(\mathbf{\Delta}_{t-1} + \eta \cdot \operatorname{sgn}(\nabla_{\mathbf{\Delta}_{t-1}} L(\mathbf{Z} + \mathbf{\Delta}_{t-1})), \tag{7}$$

where η is the step size, $\operatorname{sgn}(\cdot)$ takes the sign of each element in the input matrix, and $\Pi_{\|\mathbf{\Delta}\|_\infty \le \delta}$ projects the perturbation onto the δ-ball in the l_∞-norm.

3 METHOD

In this section, we will first investigate the vulnerability of the graph contrastive learning, then we'll spend the remaining section discussing each part of ARIEL in detail.

3.1 Vulnerability of the Graph Contrastive Learning

Many GNNs are known to be vulnerable to adversarial attacks [1, 42], so we first investigate the vulnerability of the graph neural networks trained with the contrastive learning objective in Equation (3). We generate a sequence of 60 graphs by iteratively dropping edges and masking the features. Let $G_0 = G$, for the t-th iteration, we generate G_t from G_{t-1} by randomly dropping the edges in G_{t-1} and randomly masking the unmasked features, both with probability $p = 0.03$. Since G_t is guaranteed to contain less information than G_{t-1}, G_t should be less similar to G_0 than G_{t-1}, on both the graph and node level. Denote the node embeddings of G_t as $\mathbf{H}_t$, we measure the similarity $\theta(\mathbf{H}_t[i,:], \mathbf{H}_0[i,:])$, and it is expected that the similarity decreases as the iteration goes on.

We generate the sequences on two datasets, *Amazon-Computers* and *Aamzon-Photo* [27], and the results are shown in Figure 1. At the 30-th iteration, with $0.97^{30} = 40.10\%$ edges and features left, the average similarity of the positive samples are under 0.5 on Amazon-Photo. At the 60-th iteration, with $0.97^{60} = 16.08\%$ edges and features left, the average similarity drops under 0.2 on both Amazon-Computers and Amazon-Photo. Additionally, starting from the 30-th iteration, the cosine similarity has around 0.3 standard deviation for both datasets, which indicates that a lot of nodes are actually very sensitive to the external perturbations, even if we do not add any adversarial component but just mask out some information. These results demonstrate that the current graph contrastive learning framework is not trained over enough high-quality contrastive samples and not robust to adversarial attacks.

Given this observation, we are motivated to build an adversarial graph contrastive learning framework which could improve the performance and robustness of the previous graph contrastive learning methods. The overview of our framework is shown in Figure 2.

3.2 Adversarial Training

Although most existing attack frameworks are targeted at supervised learning, it is natural to generalize these methods to contrastive learning by replacing the classification loss with the contrastive loss. The goal of the adversarial attack on contrastive learning is to maximize the contrastive loss by adding a small perturbation on the graph, which can be formulated as

$$G_{\text{adv}} = \arg\max_{G'} L_{\text{con}}(G_1, G'), \tag{8}$$

where $G' = \{\mathbf{A}', \mathbf{X}'\}$ is generated from the original graph G, and the change is constrained by the budget $\Delta_{\mathbf{A}}$ and $\Delta_{\mathbf{X}}$ as

$$\sum_{i,j} |\mathbf{A}'[i,j] - \mathbf{A}[i,j]| \le \Delta_{\mathbf{A}}, \tag{9}$$

$$\sum_{i,j} |\mathbf{X}'[i,j] - \mathbf{X}[i,j]| \le \Delta_{\mathbf{X}}. \tag{10}$$

We treat adversarial attack as one kind of data augmentations. Although we find it effective to make the adversarial attack on one or two augmented views as well, we follow the typical contrastive learning procedure as in SimCLR [4] to make the attack on the original graph in this work. Besides, it does not matter whether G_1, G_2 or G is chosen as the anchor for the adversary, each choice works in our framework and it can also be sampled as a third view. In our experiments, we use PGD attack [21] as our attack method.

We generally follow the method proposed by Xu et al. [35] to make PGD attack on the graph structure and apply the regular PGD attack method on the node features. Define the supplement of the adjacency matrix as $\bar{\mathbf{A}} = \mathbf{1}_{n\times n} - \mathbf{I}_n - \mathbf{A}$, where $\mathbf{1}_{n\times n}$ is the ones matrix of size $n \times n$. The perturbed adjacency matrix can be written as

$$\mathbf{A}_{\text{adv}} = \mathbf{A} + \mathbf{C} \circ \mathbf{L}_{\mathbf{A}}, \tag{11}$$

$$\mathbf{C} = \bar{\mathbf{A}} - \mathbf{A}, \tag{12}$$

where $\circ$ is the element-wise product, and $\mathbf{L}_{\mathbf{A}} \in \{0,1\}^{n\times n}$ with each element $\mathbf{L}_{\mathbf{A}}[i,j]$ corresponding to the modification (e.g., add, delete or no modification) of the edge between the node pair (v_i, v_j). The perturbation on $\mathbf{X}$ follows the regular PGD attack procedure and the perturbed feature matrix can be written as

$$\mathbf{X}_{\text{adv}} = \mathbf{X} + \mathbf{L}_{\mathbf{X}}, \tag{13}$$

where $\mathbf{L}_{\mathbf{X}} \in \mathbb{R}^{n\times d}$ is the perturbation on the feature matrix.

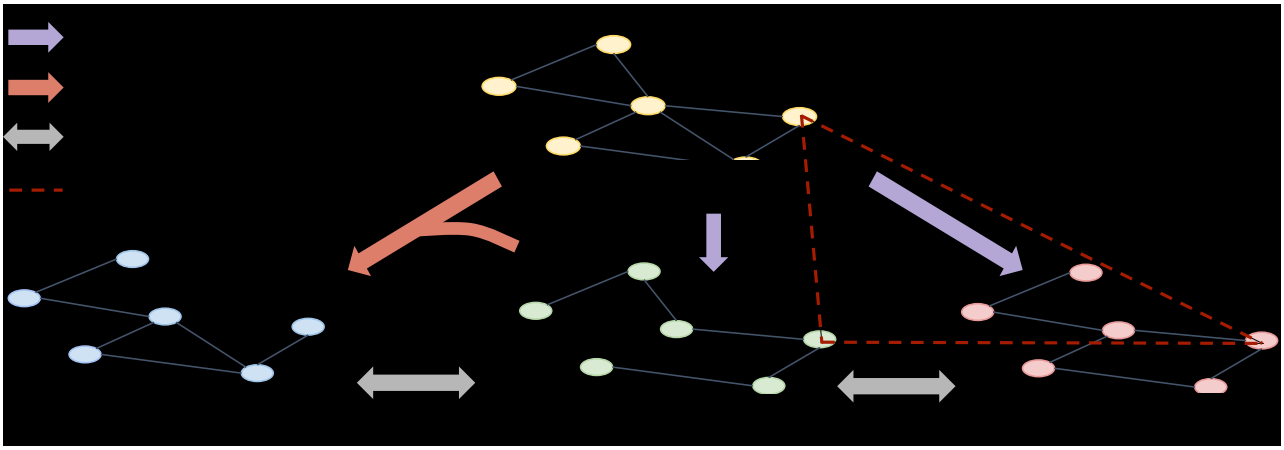

Figure 2: The overview of the proposed ARIEL framework. For each iteration, two augmented views are generated from the original graph by data augmentation (purple arrows), then an adversarial view is generated (red arrow) from the original graph by maximizing the contrastive loss against one of the augmented view. Besides, the similarities of the corresponding nodes (dashed lines) will get penalized by the information regularization if they exceed the estimated upper bound. The objective of ARIEL is to minimize the contrastive loss (grey arrows) between the augmented views, the adversarial view and the corresponding augmented view, and the information regularization. Best viewed in color.

For the ease of optimization, $\mathbf{L}_\mathbf{A}$ is relaxed to its convex hull $\tilde{\mathbf{L}}_\mathbf{A} \in [0,1]^{n\times n}$, which satisfies $\mathcal{S}_\mathbf{A} = \{\tilde{\mathbf{L}}_\mathbf{A} | \sum_{i,j} \tilde{\mathbf{L}}_\mathbf{A} \le \Delta_\mathbf{A}, \tilde{\mathbf{L}}_\mathbf{A} \in [0,1]^{n\times n}\}$. The constraint on $\mathbf{L}_\mathbf{X}$ can be written as $\mathcal{S}_\mathbf{X} = \{\mathbf{L}_\mathbf{X} | \|\mathbf{L}_\mathbf{X}\|_\infty \le \delta_\mathbf{X}, \mathbf{L}_\mathbf{X} \in \mathbb{R}^{n\times d}\}$, where we directly treat $\delta_\mathbf{X}$ as the constraint on the feature perturbation. In each iteration, we make the updates

$$\tilde{\mathbf{L}}_\mathbf{A}^{(t)} = \Pi_{\mathcal{S}_\mathbf{A}}[\tilde{\mathbf{L}}_\mathbf{A}^{(t-1)} + \alpha \cdot \mathbf{G}_\mathbf{A}^{(t)}], \tag{14}$$

$$\mathbf{L}_\mathbf{X}^{(t)} = \Pi_{\mathcal{S}_\mathbf{X}}[\mathbf{L}_\mathbf{X}^{(t-1)} + \beta \cdot \mathrm{sgn}(\mathbf{G}_\mathbf{X}^{(t)})], \tag{15}$$

where t denotes the current number of iterations, and

$$\mathbf{G}_\mathbf{A}^{(t)} = \nabla_{\tilde{\mathbf{L}}_\mathbf{A}^{(t-1)}} L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}^{(t-1)}), \tag{16}$$

$$\mathbf{G}_\mathbf{X}^{(t)} = \nabla_{\mathbf{L}_\mathbf{X}^{(t-1)}} L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}^{(t-1)}), \tag{17}$$

denote the gradients of the loss with respect to $\tilde{\mathbf{L}}_\mathbf{A}^{(t-1)}$ and $\mathbf{L}_\mathbf{X}^{(t-1)}$ respectively. Here $G_{\mathrm{adv}}^{(t-1)}$ is defined as $\{\mathbf{A}+\mathbf{C}\circ\tilde{\mathbf{L}}_\mathbf{A}^{(t-1)}, \mathbf{X}+\mathbf{L}_\mathbf{X}^{(t-1)}\}$. The projection operation $\Pi_{\mathcal{S}_\mathbf{X}}$ simply clips $\mathbf{L}_\mathbf{X}$ into the range $[-\delta_\mathbf{X}, \delta_\mathbf{X}]$. The projection operation $\Pi_{\mathcal{S}_\mathbf{A}}$ is calculated as

$$\Pi_{\mathcal{S}_\mathbf{A}}(\mathbf{Z}) = \begin{cases} P_{[0,1]}[\mathbf{Z} - \mu\mathbf{1}_{n\times n}], & \text{if } \mu > 0, \text{and} \\ & \sum_{i,j} P_{[0,1]}[\mathbf{Z} - \mu\mathbf{1}_{n\times n}] = \Delta_\mathbf{A}, \\ P_{[0,1]}[\mathbf{Z}], & \text{if } \sum_{i,j} P_{[0,1]}[\mathbf{Z}] \le \Delta_\mathbf{A}, \end{cases} \tag{18}$$

where $P_{[0,1]}[\mathbf{Z}]$ clips $\mathbf{Z}$ into the range $[0,1]$. We use the bisection method [3] to solve the equation $\sum_{i,j} P_{[0,1]}[\mathbf{Z} - \mu\mathbf{1}_{n\times n}] = \Delta_\mathbf{A}$ with respect to the dual variable μ.

To finally obtain $\mathbf{L}_\mathbf{A}$ from $\tilde{\mathbf{L}}_\mathbf{A}$, each element is independently sampled from a Bernoulli distribution as $\mathbf{L}_\mathbf{A}[i,j] \sim \mathrm{Bernoulli}(\tilde{\mathbf{L}}_\mathbf{A}[i,j])$.

3.3 Adversarial Graph Contrastive Learning

To assimilate the graph contrastive learning and adversarial training together, we treat the adversarial view G_{adv} obtained from Equation (8) as another view of the graph. We define the adversarial contrastive loss as the contrastive loss between G_1 and G_{adv}. The adversarial contrastive loss is added to the original contrastive loss in Equation (3), which becomes

$$L(G_1, G_2, G_{\mathrm{adv}}) = L_{\mathrm{con}}(G_1, G_2) + \epsilon_1 L_{\mathrm{con}}(G_1, G_{\mathrm{adv}}), \tag{19}$$

where $\epsilon_1 > 0$ is the adversarial contrastive loss coefficient. We further adopt two additional subtleties on top of this basic framework: subgraph sampling and curriculum learning. For each iteration, a subgraph G_s with a fixed size is first sampled from the original graph G, then the data augmentation and adversarial attack are both conducted on this subgraph. The subgraph sampling could avoid the gradient derivation on the whole graph, which will lead to heavy computation on a large network. And for every T epochs, the adversarial contrastive loss coefficient is multiplied by a weight γ. When $\gamma > 1$, the portion of the adversarial contrastive loss is gradually increasing and the contrastive learning becomes harder as the training goes on.

3.4 Information Regularization

Adversarial training could effectively improve the model's robustness to the perturbations, nonetheless, we find these hard training samples could impose the additional risk of training collapsing, i.e., the model will locate at a bad parameter area at the early stage of the training, assigning higher probability to a highly perturbed sample than a mildly perturbed one. In our experiment, we find this vanilla adversarial training method may fail to converge in some cases (e.g., Amazon-Photo dataset). To stabilize the training, we add one constraint termed *information regularization*.

The data processing inequality [5] states that for three random variables $\mathbf{Z}_1, \mathbf{Z}_2$ and $\mathbf{Z}_3 \in \mathbb{R}^{n\times d'}$, if they satisfy the Markov relation $\mathbf{Z_1} \rightarrow \mathbf{Z}_2 \rightarrow \mathbf{Z}_3$, then the inequality $I(\mathbf{Z}_1;\mathbf{Z}_3) \leq I(\mathbf{Z}_1;\mathbf{Z}_2)$ holds. As proved by Zhu et al. [41], since the node embeddings of two views $\mathbf{H}_1$ and $\mathbf{H}_2$ are conditionally independent given the node embeddings of the original graph $\mathbf{H}$, they also satisfy the Markov relation with $\mathbf{H}_1 \rightarrow \mathbf{H} \rightarrow \mathbf{H}_2$, and vice versa. Therefore, we can derive the following properties over their mutual information

$$I(\mathbf{H}_1;\mathbf{H}_2) \leq I(\mathbf{H};\mathbf{H}_1), \tag{20}$$

$$I(\mathbf{H}_1;\mathbf{H}_2) \leq I(\mathbf{H};\mathbf{H}_2). \tag{21}$$

In fact, this inequality holds on each node. A sketch of the proof is that the embedding of each node v_i is determined by all the nodes from its l-hop neighbourhood if a l-layer GNN is used as the encoder, and this subgraph composed of its l-hop neighbourhood also satisfies the Markov relation. Therefore, we can derive the more strict inequalities

$$I(\mathbf{H}_1[i,:];\mathbf{H}_2[i,:]) \leq I(\mathbf{H}[i,:];\mathbf{H}_1[i,:]), \tag{22}$$

$$I(\mathbf{H}_1[i,:];\mathbf{H}_2[i,:]) \leq I(\mathbf{H}[i,:];\mathbf{H}_2[i,:]). \tag{23}$$

Since $-L_{\text{con}}(G_1, G_2)$ is only a lower-bound of the mutual information, directly applying the above constraints is hard, we only consider the constraints on the density ratio. Using the Markov relation for each node, we give the following theorem

Theorem 3.1. *For two graph views G_1 and G_2 independently transformed from the graph G, the density ratio of their node embeddings $\mathbf{H}_1$ and $\mathbf{H}_2$ should satisfy $g(\mathbf{H}_2[i,:],\mathbf{H}_1[i,:]) \leq g(\mathbf{H}_2[i,:],\mathbf{H}[i,:])$ and $g(\mathbf{H}_1[i,:],\mathbf{H}_2[i,:]) \leq g(\mathbf{H}_1[i,:],\mathbf{H}[i,:])$, where $\mathbf{H}$ is the node embeddings of the original graph.*

Proof. Following the Markov relation of each node, we get

$$p(\mathbf{H}_2[i,:]|\mathbf{H}_1[i,:]) = p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:])p(\mathbf{H}[i,:]|\mathbf{H}_1[i,:]) \tag{24}$$

$$\leq p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:]), \tag{25}$$

and consequently

$$\frac{p(\mathbf{H}_2[i,:]|\mathbf{H}_1[i,:])}{p(\mathbf{H}_2[i,:])} \leq \frac{p(\mathbf{H}_2[i,:]|\mathbf{H}[i,:])}{p(\mathbf{H}_2[i,:])}. \tag{26}$$

Since $g(\mathbf{a},\mathbf{b}) \propto \frac{p(\mathbf{a}|\mathbf{b})}{p(\mathbf{a})}$, we get $g(\mathbf{H}_2[i,:],\mathbf{H}_1[i,:]) \leq g(\mathbf{H}_2[i,:],\mathbf{H}[i,:])$. The similar proof applies for the other inequality. □

Note that $g(\cdot,\cdot)$ is symmetric for the two inputs, we thus get two upper bounds for $g(\mathbf{H}_1[i,:],\mathbf{H}_2[i,:])$. According to the previous definition, $g(\mathbf{a},\mathbf{b}) = e^{\theta(\mathbf{a},\mathbf{b})/\tau}$, we can simply replace $g(\cdot,\cdot)$ with $\theta(\cdot,\cdot)$ in the inequalities, then we combine these two upper bounds into one

$$2\cdot\theta(\mathbf{H}_1[i,:],\mathbf{H}_2[i,:]) \leq \theta(\mathbf{H}_2[i,:],\mathbf{H}[i,:]) + \theta(\mathbf{H}_1[i,:],\mathbf{H}[i,:]). \tag{27}$$

This bound intuitively requires the similarity between $\mathbf{H}_1[i,:]$ and $\mathbf{H}_2[i,:]$ to be smaller than the similarity between $\mathbf{H}[i,:]$ and $\mathbf{H}_1[i,:]$ or $\mathbf{H}_2[i,:]$. Therefore, we define the following information regularization to penalize the higher probability of a less similar pair

$$d_i = 2\cdot\theta(\mathbf{H}_1[i,:],\mathbf{H}_2[i,:]) - (\theta(\mathbf{H}_2[i,:],\mathbf{H}[i,:]) + \theta(\mathbf{H}_1[i,:],\mathbf{H}[i,:])) \tag{28}$$

$$L_I(G_1,G_2,G) = \frac{1}{n}\sum_{i=1}^{n}\max\{d_i,0\}. \tag{29}$$

Specifically, the information regularization could be defined over any three graphs that satisfy the Markov relation, but for our framework, to save the memory and time complexity, we avoid additional sampling and directly ground the information regularization on the existing graphs.

The final loss of ArieL can be written as

$$L(G_1,G_2,G_{\text{adv}}) = L_{\text{con}}(G_1,G_2) + \epsilon_1 L_{\text{con}}(G_1,G_{\text{adv}}) + \epsilon_2 L_I(G_1,G_2,G), \tag{30}$$

where $\epsilon_2 > 0$ controls the strength of the information regularization.

The entire algorithm of ArieL is summarized in Algorithm 1.

Algorithm 1 Algorithm of ArieL

Input data: Graph $G = (\mathbf{A},\mathbf{X})$
Input parameters: α, β, $\Delta_{\mathbf{A}}$, $\delta_{\mathbf{X}}$, ϵ_1, ϵ_2, γ and T
Randomly initialize the graph encoder f
for iteration $k = 0, 1, \cdots$ **do**
  Sample a subgraph G_s from G
  Generate two views G_1 and G_2 from G_s
  Generate the adversarial view G_{adv} according to Equations (15), (14)
  Update model f to minimize $L(G_1,G_2,G_{\text{adv}})$ in Equation (30)
  if $(k+1) \mod T = 0$ **then**
    Update $\epsilon_1 \leftarrow \gamma * \epsilon_1$
  end if
end for
return: Node embedding matrix $\mathbf{H} = f(\mathbf{A},\mathbf{X})$

4 EXPERIMENTS

In this section, we conduct empirical evaluations, which are designed to answer the following three questions:

- RQ1. How effective is the proposed ArieL in comparison with previous graph contrastive learning methods on the node classification task?
- RQ2. To what extent does ArieL gain robustness over the attacked graph?
- RQ3. How does each part of ArieL contribute to its performance?

We evaluate our method with the node classification task on both the real-world graphs and attacked graphs. The node embeddings are first learnt by the proposed ArieL algorithm, then the embeddings are fixed to perform the node classification with a logistic regression classifier trained over it. All our experiments are conducted on the NVIDIA Tesla V100S GPU with 32G memory.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
Amazon-Computers	13,752	245,861	767	10
Amazon-Photo	7,650	119,081	745	8
Coauthor-CS	18,333	81,894	6,805	15
Coauthor-Physics	34,493	247,962	8,415	5

Table 1: Datasets statistics, number of nodes, edges, node feature dimensions and classes are listed.

4.1 Experimental Setup

4.1.1 Datasets. We use six datasets for the evaluation, including *Cora*, *CiteSeer*, *Amazon-Computers*, *Amazon-Photo*, *Coauthor-CS* and *Coauthor-Physics*. A summary of the dataset statistics is in Table 1.

4.1.2 Baselines. We consider six graph contrastive learning methods, including DeepWalk [24], DGI [32], GMI [23], MVGRL [7], GRACE [40] and GCA [41]. Since DeepWalk only generates the embeddings for the graph topology, we concatenate the node features to the generated embeddings for the evaluation so that the final embeddings can incorporate both the topology and features information. Besides, we also compare our method with two supervised methods Graph Convolutional Network (GCN) [17] and Graph Attention Network (GAT) [31].

4.1.3 Evaluation protocol. For each dataset, we randomly select 10% nodes for training, 10% nodes for validation and the remaining for testing. For contrastive learning methods, a logistic regression classifier is trained to do the node classification over the node embeddings. The accuracy is used as the evaluation metric.

Besides testing on the original, clean graphs, we also evaluate our method on the attacked graphs. We use Metattack [43] to perform the poisoning attack. For ARIEL we use the hyperparameters of the best models we obtain on the clean graphs for evaluation. For GCA, we report the performance for its three variants, GCA-DE, GCA-PR and GCA-EV, which corresponds to the adoption of degree, eigenvector and PageRank [15, 22] centrality measures, in our main results and use the best variant on each dataset for the evaluation on the attacked graphs.

4.1.4 Hyperparameters. For ARIEL we use the same parameters and design choices for network architecture, optimizer and training scheme as in GRACE and GCA on each dataset. However, we find GCA not behave well on Cora with a significant performance drop, so we re-search the parameters for GCA on Cora separately and use a different temperature for it. Other contrastive learning specific parameters are kept the same over GRACE, GCA and ARIEL.

All GNN based baselines use a two-layer GCN as the encoder. For each method, we compare its default hyperparameters and the ones used by ARIEL, and use the hyperparameters leading to the better performance. Other algorithm-specific hyperparameters all respect the default setting in its official implementation.

Other hyperparameters of ARIEL are summarized in the Appendix.

4.2 Main Results

The comparison results of node classification on all six datasets are summarized in Table 2.

Our method ARIEL outperforms baselines over all datasets except on Cora, with only 0.11% lower in accuracy than MVGRL. It can be seen that the state of the art method GCA does not bear significant improvements over previous methods. In contrast, ARIEL can achieve the consistent improvements over GRACE and GCA on all datasets, especially on Amazon-Computers with almost 3% gain.

Besides, we find MVGRL a solid baseline whose performance is close to or even better than GCA on these datasets. It achieves the highest score on Cora and the second-highest on Amazon-Computers and Amazon-Photo. However, it does not behave well on CiteSeer, where GCA can effectively increase the score of GRACE. To sum up, previous modifications over the grounded frameworks are mostly based on specific knowledge, for example, MVGRL introduces the diffusion matrix to DGI and GCA defines the importance on the edges and features, and they cannot consistently take effect on all datasets. However, ARIEL uses the adversarial attack to automatically construct the high-quality contrastive samples, and achieves more stable performance improvements.

Finally, in comparison with the supervised methods, ARIEL also achieves a clear advantage over all of them. Although it would be premature to conclude that ARIEL is more powerful than these supervised methods since they are usually tested under the specific training-testing split, these results do demonstrate that ARIEL can indeed generate highly expressive node embeddings for the node classification task, which can achieve comparable performance to the supervised methods.

4.3 Results under Attack

The results on attacked graphs are summarized in Table 3. Specifically, we evaluate all these methods on the attacked subgraph of Amazon-Computers, Amazon-Photo, Coauthor-CS and Coauthor-Physics, so their results are not directly comparable to the results in Table 2.

It can be seen that although some baselines are robust on specific datasets, for example, MVGRL on Cora, GMI on CiteSeer, and GCA on Coauthor-CS and Coauthor-Physics, they fail to achieve the consistent robustness over all datasets. Although GCA indeed makes GRACE more robust, it is still vulnerable on Cora, CiteSeer and Amazon-Computers, with more than 3% lower than ARIEL in the final accuracy.

Basically, MVGRL and GCA can improve the robustness of their respective grounded frameworks over different datasets, but we find this kind of improvement relatively minor. Instead, ARIEL has more significant improvements and greatly increases robustness.

4.4 Ablation Study

For this section, we first set ϵ_2 as 0 and investigate the role of adversarial contrastive loss. The adversarial contrastive loss coefficient ϵ_1 controls the portion of the adversarial contrastive loss in the final loss. When $\epsilon_1 = 0$, the final loss reduces to the regular contrastive loss in Equation (3). To explore the effect of the adversarial contrastive loss, we fix other parameters in our best models on Cora

Method	Cora	CiteSeer	Amazon-Computers	Amazon-Photo	Coauthor-CS	Coauthor-Physics
GCN	84.14 ± 0.68	69.02 ± 0.94	88.03 ± 1.41	92.65 ± 0.71	92.77 ± 0.19	95.76 ± 0.11
GAT	83.18 ± 1.17	69.48 ± 1.04	85.52 ± 2.05	91.35 ± 1.70	90.47 ± 0.35	94.82 ± 0.21
DeepWalk+features	79.82 ± 0.85	67.14 ± 0.81	86.23 ± 0.37	90.45 ± 0.45	85.02 ± 0.44	94.57 ± 0.20
DGI	84.24 ± 0.75	69.12 ± 1.29	88.78 ± 0.43	92.57 ± 0.23	92.26 ± 0.12	95.38 ± 0.07
GMI	82.43 ± 0.90	70.14 ± 1.00	83.57 ± 0.40	88.04 ± 0.59	OOM	OOM
MVGRL	**84.39 ± 0.77**	69.85 ± 1.54	89.02 ± 0.21	92.92 ± 0.25	92.22 ± 0.22	95.49 ± 0.17
GRACE	83.40 ± 1.08	69.47 ± 1.36	87.77 ± 0.34	92.62 ± 0.25	93.06 ± 0.08	95.64 ± 0.08
GCA-DE	82.57 ± 0.87	72.11 ± 0.98	88.10 ± 0.33	92.87 ± 0.27	93.08 ± 0.18	95.62 ± 0.13
GCA-PR	82.54 ± 0.87	72.16 ± 0.73	88.18 ± 0.39	92.85 ± 0.34	93.09 ± 0.15	95.58 ± 0.12
GCA-EV	81.80 ± 0.92	67.07 ± 0.79	87.95 ± 0.43	92.63 ± 0.33	93.06 ± 0.14	95.64 ± 0.08
ARIEL	84.28 ± 0.96	**72.74 ± 1.10**	**91.13 ± 0.34**	**94.01 ± 0.23**	**93.83 ± 0.14**	**95.98 ± 0.05**

Table 2: Node classification accuracy in percentage on six real-world datasets. We bold the results with the best mean accuracy. The methods above the line are the supervised ones, and the ones below the line are unsupervised. OOM stands for Out-of-Memory on our 32G GPUs.

Method	Cora	CiteSeer	Amazon-Computers	Amazon-Photos	Coauthor-CS	Coauthor-Physics
GCN	80.03 ± 0.91	62.98 ± 1.20	84.10 ± 1.05	91.72 ± 0.94	80.32 ± 0.59	87.47 ± 0.38
GAT	79.49 ± 1.29	63.30 ± 1.11	81.60 ± 1.59	90.66 ± 1.62	77.75 ± 0.80	86.65 ± 0.41
DeepWalk+features	74.12 ± 1.02	63.20 ± 0.80	79.08 ± 0.67	88.06 ± 0.41	49.30 ± 1.23	79.26 ± 1.38
DGI	80.84 ± 0.82	64.25 ± 0.96	83.36 ± 0.55	91.27 ± 0.29	78.73 ± 0.50	85.88 ± 0.37
GMI	79.17 ± 0.76	65.37 ± 1.03	77.42 ± 0.59	89.44 ± 0.47	80.92 ± 0.64	87.72 ± 0.45
MVGRL	**80.90 ± 0.75**	64.81 ± 1.53	83.76 ± 0.69	91.76 ± 0.44	79.49 ± 0.75	86.98 ± 0.61
GRACE	78.55 ± 0.81	63.17 ± 1.81	84.74 ± 1.13	91.26 ± 0.37	80.61 ± 0.63	85.71 ± 0.38
GCA	76.79 ± 0.97	64.89 ± 1.33	85.05 ± 0.51	91.71 ± 0.34	82.72 ± 0.58	89.00 ± 0.31
ARIEL	80.33 ± 1.25	**69.13 ± 0.94**	**88.61 ± 0.46**	**92.99 ± 0.21**	**84.43 ± 0.59**	**89.09 ± 0.31**

Table 3: Node classification accuracy in percentage on the graphs under Metattack, where subgraphs of Amazon-Computers, Amazon-Photo, Coauthor-CS and Coauthor-Physics are used for attack and their results are not directly comparable to those in Table 2. We bold the results with the best mean accuracy. GCA is evaluated on its best variant on each clean graph.

and CiteSeer, and gradually increase ϵ_1 from 0 to 2. The changes of the final performance are shown in Figure 3.

The dashed line represents the performance of GRACE with subgraph sampling, i.e., $\epsilon_1 = 0$. Although there exist some variations, ARIEL is always above the baseline under a positive ϵ_1 with around 2% improvement. The subgraph sampling trick may sometimes help the model, for example, it improves GRACE without subgraph sampling by 1% on CiteSeer, but it could be detrimental as well, such as on Cora. This is understandable since subgraph sampling can simultaneously enrich the data augmentation and lessen the number of negative samples, both critical to the contrastive learning. While for the adversarial contrastive loss, it has a stable and significant improvement on GRACE with subgraph sampling, which demonstrates that the performance improvement of ARIEL mainly stems from the adversarial loss rather than the subgraph sampling.

Next, we fix all other parameters and check the behavior of ϵ_2. Information regularization is mainly designed to stabilize the training of ARIEL. We find ARIEL would experience the collapsing at the early training stage and the information regularization could mitigate this issue. We choose the best run on Amazon-Photo, where the collapsing frequently occurs, and similar to before, we gradually increase ϵ_2 from 0 to 2, the results are shown in Figure 4 (left). As

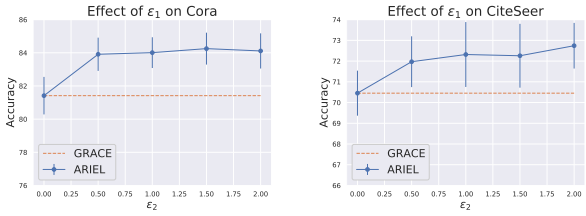

Figure 3: Effect of adversarial contrastive loss coefficient ϵ_1 on Cora and CiteSeer. The dashed line represents the performance of GRACE with subgraph sampling.

can be seen, without using the information regularization, ARIEL could collapse without learning anything, while setting ϵ_2 greater than 0 can effectively avoid this situation. To further illustrate this, we draw the training curve of the regular contrastive loss in Figure 4 (right), for the best ARIEL model on Amazon-Photo and the same model by simply removing the information regularization. Without information regularization, the model could get stuck in a bad parameter and fail to converge, while information regularization can resolve this issue.

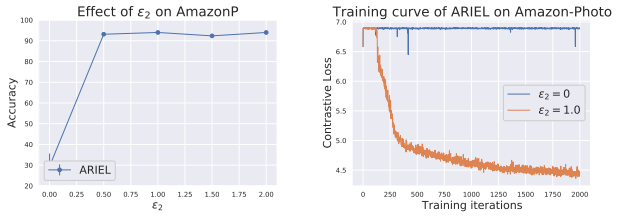

Figure 4: Effect of information regularization on Amazon-Photo. Left figure shows the model performance under different ϵ_2 and the right figure plots the training curve of ARIEL under $\epsilon_2 = 0$ and $\epsilon_2 = 1.0$.

5 RELATED WORK

In this section, we review the related work in the following three categories: graph constrastive learning, adversarial attack on graphs and adversarial contrastive learning.

5.1 Graph Contrastive Learning

Contrastive learning is known for its simplicity and strong expressivity. Traditional methods ground the contrastive samples on the node proximity in the graph, such as DeepWalk [24] and node2vec [6], which use random walks to generate the node sequences and approximate the co-occurrence probability of node pairs. However, these methods can only learn the embeddings for the graph structures but regardless of the node features.

GNNs [17, 31] can easily capture the local proximity and node features. To further improve the performance, Information Maximization (InfoMax) principle [19] has been introduced. DGI [32] is adapted from Deep InfoMax [9] to maximize the mutual information between the local and global features. It generates the negative samples with a corrupted graph and contrast the node embeddings with the original graph embedding and the corrupted one. Based on a similar idea, GMI [23] generalizes the concept of mutual information to the graph domain by separately defining the mutual information on the features and edges. The other followup work of DGI, MVGRL [7], maximizes the mutual information between the first-order neighbours and graph-diffusion. HDMI [13] introduces high-order mutual information to consider both intrinsic and extrinsic training signals. However, mutual inforamtion based methods generate the corrupted graphs by simply randomly shuffling the node features. Recent methods exploit the graph topology and features to generate better augmented graphs. GCC [25] adopts a random-walk based strategy to generate different views of the context graph for a node, but it ignores the augmentation of the feature level. GCA [40], instead, considers the data augmentation from both the topology and feature level, and introduces the adaptive augmentation by considering the importance of each edge and feature dimension. Unlike the above methods which constructs the data augmentation samples based on domain knowledge, ARIEL uses adversarial attack to construct the view that maximizes the contrastive loss, which is more informative with broader applicability.

5.2 Adversarial Attack on Graphs

Deep learning methods on graphs are known vulnerable to the adversarial attack. As shown by Bojchevski et al. [1], both random-walk based methods and GNN based methods could be attacked by flipping a small portion of edges. Xu et al. [35] proposes PGD attack and min-max attack on the graph structure from the optimization perspective. NETTACK [42] is the first to attack GNNs using both structure attack and feature attack, causing the significant performance drop of GNNs on the benchmarks. After that, Metattack [43] formulates the poisoning attack of GNNs as a meta learning problem and achieves remarkable performance by only perturbing the graph structure. Node Injection Poisoning Attacks (NIPA) uses a hierarchical reinforcement learning approach to sequentially manipulate the labels and links of the injected nodes. Recently, InfMax [20] formulates the adversarial attack on GNNs as an influence maximization problem.

5.3 Adversarial Contrastive Learning

The concept of adversarial contrastive learning is first proposed on visual domains [10, 12, 16]. All these works propose the similar idea to use the adversarial sample as a form of data augmentation in contrastive learning and it can bring a better downstream task performance and higher robustness. Recently, AD-GCL [28] formulates adversarial graph contrastive learning in a min-max form and uses a parameterized network for edge dropping. However, AD-GCL is designed for the graph classification task instead and it does not explore the robustness of graph contrastive learning. Moreover, both visual adversarial contrastive learning methods and AD-GCL deals with independent instances, e.g., independent images or graphs, while the instances studied by ARIEL has interdependence, which makes the scalability a challenging issue.

6 CONCLUSION

In this paper, we propose a simple yet effective framework for graph contrastive learning by introducing an adversarial view and we stabilize it through the information regularization. It consistently outperforms the state of the art graph contrastive learning methods in the node classification task and exhibits a higher degree of robustness to the adversarial attack. Our framework is not limited to the graph contrastive learning framework we build on in this paper, and it can be naturally extended to other graph contrastive learning methods as well. In the future, we plan to further investigate (1) the adversarial attack on graph contrastive learning (2) the integration of graph contrastive learning and supervised methods.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under grant No. 1947135, DARPA HR001121C0165, Department of Homeland Security under Grant Award Number 17STQAC00001-03-03, NIFA award 2020-67021-32799 and Army Research Office (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Aleksandar Bojchevski and Stephan Günnemann. 2019. Adversarial Attacks on Node Embeddings via Graph Poisoning. arXiv:1809.01093 [cs.LG]

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc., 2787–2795. https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

[3] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press. https://doi.org/10.1017/CBO9780511804441

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709 [cs.LG]

[5] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. arXiv:1607.00653 [cs.SI]

[7] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. arXiv:2006.05582 [cs.LG]

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. arXiv:1911.05722 [cs.CV]

[9] 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. arXiv:1808.06670 [stat.ML]

[10] Chih-Hui Ho and Nuno Vasconcelos. 2020. Contrastive Learning with Adversarial Examples. arXiv:2010.12050 [cs.CV]

[11] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[12] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. 2020. Robust Pre-Training by Adversarial Contrastive Learning. arXiv:2010.13337 [cs.CV]

[13] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*. 2414–2424.

[14] Nikola Jovanović, Zhao Meng, Lukas Faber, and Roger Wattenhofer. 2021. Towards Robust Graph Contrastive Learning. arXiv:2102.13085 [cs.LG]

[15] Jian Kang, Meijia Wang, Nan Cao, Yinglong Xia, Wei Fan, and Hanghang Tong. 2018. Aurora: Auditing pagerank on large graphs. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 713–722.

[16] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. 2020. Adversarial Self-Supervised Contrastive Learning. arXiv:2006.07589 [cs.LG]

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG]

[18] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2181–2187.

[19] Ralph Linsker. 1988. Self-organization in a perceptual network. *Computer* 21, 3 (1988), 105–117.

[20] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2021. Adversarial Attack on Graph Neural Networks as An Influence Maximization Problem. arXiv:2106.10785 [cs.LG]

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [stat.ML]

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/ Previous number = SIDL-WP-1999-0120.

[23] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. arXiv:2002.01169 [cs.LG]

[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, New York, USA) *(KDD '14)*. ACM, New York, NY, USA, 701–710. https://doi.org/10.1145/2623330.2623732

[25] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.

[26] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive Learning with Hard Negative Samples. In *International Conference on Learning Representations*. https://openreview.net/forum?id=CR1XOQ0UTh-

[27] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Pitfalls of Graph Neural Network Evaluation. arXiv:1811.05868 [cs.LG]

[28] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021. Adversarial Graph Augmentation to Improve Graph Contrastive Learning. arXiv:2106.05819 [cs.LG]

[29] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*. IEEE, 613–622.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748 [cs.LG]

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML]

[32] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. arXiv:1809.10341 [stat.ML]

[33] Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. 2018. Acekg: A large-scale knowledge graph for academic data mining. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1487–1490.

[34] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.

[35] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. arXiv:1906.04214 [cs.LG]

[36] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021. Dynamic Knowledge Graph Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4564–4572.

[37] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. arXiv:1603.08861 [cs.LG]

[38] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. arXiv:2010.13902 [cs.LG]

[39] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2020. Data Augmentation for Graph Neural Networks. arXiv:2006.06830 [cs.LG]

[40] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. arXiv:2006.04131 [cs.LG]

[41] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. *Proceedings of the Web Conference 2021* (Apr 2021). https://doi.org/10.1145/3442381.3449802

[42] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial Attacks on Neural Networks for Graph Data. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Jul 2018). https://doi.org/10.1145/3219819.3220078

[43] Daniel Zügner and Stephan Günnemann. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bylnx209YX

A ADDITIONAL EXPERIMENTAL SETUP DETAILS

In this section, we describe some experimental setup details not covered in our main text.

The six datasets used in this paper, including *Cora*, *CiteSeer*, *Amazon-Computers*, *Amazon-Photo*, *Coauthor-CS* and *Coauthor-Physics*, are from Pytorch Geometric [1]. Cora and CiteSeer [37] are citation networks, where nodes represent documents and edges correspond to citations. Amazon-Computers and Amazon-Photo [27] are extracted from the Amazon co-purchase graph. In these graphs, nodes are the goods and they are connected by an edge if they are frequently bought together. Coauthor-CS and Coauthor-Physics are [27] the co-authorship graphs, where each node is an author and the edge indicates the co-authorship on a paper.

For the evaluation, different from the logistic regression implementation used by DGI [32], we use the implementation from scikit-learn [2] directly since it has a better performance.

We search each method over 6 different random seeds, including 5 random seeds from our own and the best random seed of GCA on each datset. For each seed, we evaluate the method on 20 random training-validation-testing dataset splits, and report the mean and the standard deviation of the accuracy on the best seed. Specifically, for the supervised learning methods, we abandon the existing splits, for example on Cora and CiteSeer, but instead doing a random split before the training and report the results over 20 splits.

On the attacked graphs, since Metattack [43] is targeted at graph structure only and computationally inefficient on the large graphs, we first randomly sample a subgraph of 5000 nodes if the number of nodes in the original graph is greater than 5000, then we randomly mask out 20% features, and finally use Metattack to purterb 20% edges to generate the final attacked graph.

B HYPERPARAMETERS OF ARIEL

In this section, we give the details about the hyperparameter searching for ARIEL in our experiment, they are summarized as follows:

- Adversarial contrastive loss coefficient ϵ_1 and information regularization strength ϵ_2. We search them over $\{0.5, 1, 1.5, 2\}$ and use the one with the best performance on the validation set of each dataset. Specifically, we first fix ϵ_2 as 0 and decide the optimal value for other parameters, then we search ϵ_2 on top of the model with all other hyperparameters fixed.
- Number of attack steps and perturbation constraint. These parameters are fixed on all datasets, with the number of attack steps 5, edge perturbation constraint $\Delta_{\mathbf{A}} = 0.1 \sum_{i,j} \mathbf{A}[i, j]$ and feature perturbation constraint $\delta_{\mathbf{X}} = 0.5$.
- Curriculum learning weight γ and change period T. In our experiments, we simply fix $\gamma = 1.1$ and $T = 20$.
- Graph perturbation rate α and feature perturbation rate β. We search both of them over $\{0.001, 0.01, 0.1\}$ and take the best one on the validation set of each dataset.
- Subgraph size. We keep the subgraph size 500 for ARIEL on all datasets. Although we find increasing the subgraph size benefits ARIEL on larger graphs, using this fixed size still ensures a solid performance of ARIEL.

C TRAINING ANALYSIS

Here we compare the training of ARIEL to other methods on our NVIDIA Tesla V100S GPU with 32G memory.

Adversarial attacks on graphs tend to be highly computationally expensive since the attack requires the gradient calculation over the entire adjacency matrix, which is of size $O(n^2)$. For ARIEL, we resolve this bottleneck with subgraph sampling and empirically show that the adversarial training on the subgraph still yield the significant improvements, without increasing the number of training iterations. In our experiments, we find GMI the most memory inefficient, which cannot be trained on Coauthor-CS and Coauthor-Physics. For DGI, MVGRL, GRACE and GCA, the training of them also amounts to 30G GPU memory on Coauthor-Physics while the training of ARIEL requires no more than 8G GPU memory. In terms of the training time, DGI and MVGRL are the fastest to converge, but it takes MVGRL long time to compute the diffusion matrix on large graphs. ARIEL is slower than GRACE and GCA on Cora and CiteSeer, but it is faster on the large graphs like Coauthor-CS and Coauthor-Physics, with the training time for each iteration invariant to the graph size due to the subgraph sampling. On the largest graph Coauthor-Physics, each iteration takes GRACE 0.875 second and GCA 1.264 seconds, while it only takes ARIEL 0.082 second. This demonstrates that ARIEL has even better scalability than GRACE and GCA.

[1] All the datasets are from Pytorch Geometric 1.6.3: https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html

[2] https://scikit-learn.org/stable/