Adversarial Contrastive Graph Masked AutoEncoder Against Graph Structure and Feature Dual Attacks

Weixuan Shen[1], **Xiaobo Shen**[1*], **Shirui Pan**[2]

[1]Nanjing University of Science and Technology, Nanjing, China
[2]Griffith University, Gold Coast, Australia
wxshen@njust.edu.cn, njust.shenxiaobo@gmail.com, s.pan@grifith.edu.au

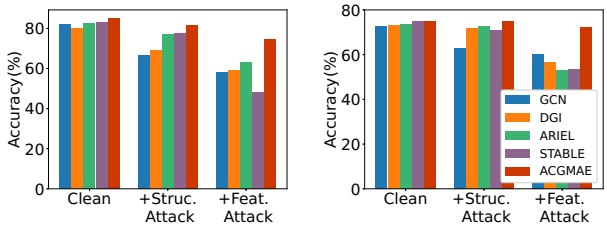
Abstract

Graph Neural Networks (GNNs) have been shown vulnerable to graph adversarial attacks. Current robust graph representation learning methods mainly defend against graph structure attack, and improve performance of GNNs. However, node features in graph can also be easily attacked in reality. The joint defense on graph structure and feature dual attacks remains challenging yet less studied. To fulfill this gap, we propose **A**dversarial **C**ontrastive **G**raph **M**asked **A**uto**E**ncoder (ACGMAE) to defend against graph structure and feature dual attacks. ACGMAE employs adversarial feature masking for reconstructing node features to mitigate the influence of feature attack. Additionally, ACGMAE employs contrastive learning on kNN graph and attacked graph, considering neighbor nodes as positive samples. By calculating the probabilities of these neighbors being true positive, ACGMAE effectively reduces the influence of adversarial edges. Extensive experiments on node classification and clustering tasks demonstrate the effectiveness of the proposed ACGMAE, especially under graph structure and feature dual attacks.

Introduction

Graphs are ubiquitous data structures that exist in many real-world scenarios, e.g., social networks, knowledge graphs, and chemical molecules. It is crucial to learn effective graph embeddings and apply them to downstream tasks. Graph Neural Networks (GNNs) have developed rapidly and made breakthroughs in many graph analysis tasks, e.g., node classification (Kipf and Welling 2017), link prediction (Zhang and Chen 2018), and graph classification (Han et al. 2022). The core of GNNs lies in message passing mechanism (Gilmer et al. 2017), where GNNs regard node features and hidden representations as messages, and propagate them through edges.

Despite achieving satisfactory performance, most existing GNNs assume that graphs are clean, which usually does not hold in reality. Some studies (Dai et al. 2018; Sun et al. 2022) have underscored the vulnerability of GNNs to adversarial structural attacks, and showed that subtle perturbations on graph structure, primarily in the form of edge additions or deletions, can severely degrade the performance of downstream tasks. The lack of robustness of GNNs may lead to serious consequences in some critical real-world applications, e.g., financial transactions, fraud detection. Therefore, some studies focus on improving the robustness of GNNs against graph structural attacks by designing new architecture (Jin et al. 2021; Geisler, Zügner, and Günnemann 2020) or refining graph structure (Jin et al. 2020; Xu et al. 2021b). Figure 1 reports the accuracies of node classification of five representative graph embedding methods on two benchmark datasets, i.e., Cora, Citeseer. The performances of two conventional GNNs, i.e., GCN (Kipf and Welling 2017) and DGI (Velickovic et al. 2019) on attacked graphs are obviously degraded in comparison to those on clean graphs, and two robust graph embedding methods, i.e., STABLE (Li et al. 2022), ARIEL (Feng et al. 2022) exhibit notably less degradation under graph structure attack. As label is usually expensive to obtain, some recent efforts (Feng et al. 2022; Li et al. 2022) have been made towards unsupervised robust graph embedding learning.



(a) Cora (b) Citeseer

Figure 1: Performance of representative graph embedding methods under incremental structure attack (25% *metattack*) and feature attack (50% normally distributed noise).

In addition to graph structure, node features are frequently attacked as well. In financial networks, for example, attackers may manipulate customers' transaction histories and credit scores, thereby posing significant challenges for fraud detection systems to accurately identify potential fraudulent activities. Existing robust graph representation learning (GRL) methods mainly focus on defending against graph structural attack, and employ properties of node features (Jin et al. 2020; Li et al. 2022), e.g., feature smoothing to reduce

*Corresponding author.

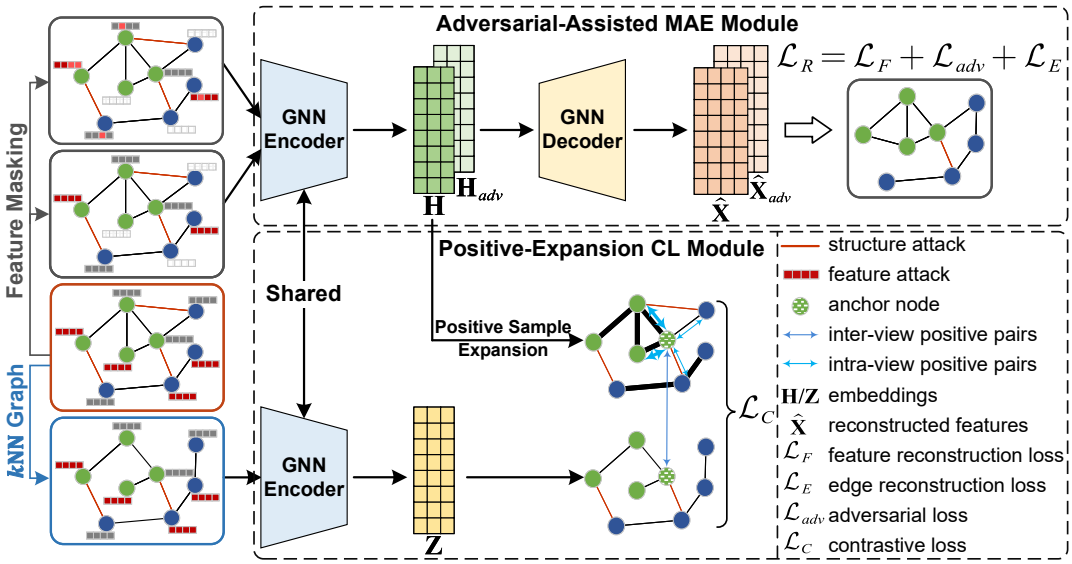


Figure 2: The overview of the proposed ACGMAE. It comprises adversarial-assisted MAE and positive-expension CL modules. The MAE module performs adversarial training to assist feature reconstruction. The CL module calculates probabilities of neighbor nodes being true positive, and performs contrastive learning on kNN graph and attacked graph. The thickness of the edges indicates probability of two neighbor nodes.

the impact of adversarial edges. Their inability to defend against graph feature attack leads to notable degradation in performance. As evidenced in Figure 1, both STABLE and ARIEL exhibit significant performance degradation under feature attack. Therefore, it still remains a research gap in developing robust GRL methods that can effectively defend against dual attacks of graph structure and feature.

To address the above issue, this paper proposes Adversarial Contrastive Graph Masked AutoEncoder (ACGMAE) to learn robust graph embeddings. The proposed ACGMAE comprises two modules, including adversarial-assisted masked autoencoder (MAE) and positive-expansion contrastive learning (CL) modules, as shown in Figure 2. The proposed ACGMAE employs adversarial masked feature and topology reconstruction, and performs contrastive learning using expended positive samples. Figure 1 empirically demonstrates the superior defensive capability of the proposed ACGMAE on defending against graph structure and feature dual attacks, significantly outperforming existing GRL methods. The main contributions of this work are as follows:

- We propose a new unsupervised GRL method called Adversarial Contrastive Graph Masked AutoEncoder (ACGMAE) to learn robust graph embeddings. To our knowledge, ACGMAE is among the first attempts to defend against dual attacks on both graph structure and feature without relying on label supervision.
- We introduce adversarial feature masking to simulate feature attack during feature reconstruction, effectively reducing sensitivity to feature attack. Moreover, we design the positive-expansion contrastive learning based on weight allocation, enabling the identification of normal and adversarial edges.
- The proposed method is verified on node classification and clustering under various attacks, and empirical results demonstrate that the proposed method outperforms the state-of-the-arts under graph structure and feature dual attacks.

Related Work

Graph Representation Learning

GNNs are powerful models for learning graph representations by capturing complex dependencies within graphs. GCN (Kipf and Welling 2017) is a popular GNN model that utilizes spectral graph convolution to gather information from neighbor nodes. Another notable GNN is the Graph Attention Network (GAT) (Veličković et al. 2017) which incorporates attention mechanisms to focus on relevant neighborhood information. One challenge faced by GNNs is their reliance on labeled data for training, which can be costly and difficult to obtain in real-world applications. To address this challenge, researchers have explored unsupervised methods, generally categorized into generative learning and contrastive learning.

In generative learning, Graph AutoEncoders (GAEs) stand as the predominant models, utilizing an "encoding-decoding" paradigm to recover graph properties. VGAE (Kipf and Welling 2016) leverages a simple 2-layer GCN as the encoder and dot-product as the decoder to get a reconstructed topology graph. RGVAE (Ma, Chen, and Xiao 2018) further imposes validity constraints on the graph variational autoencoder to regularize the output distribution of the decoder. Inspired by the success of masked autoencoders (MAEs) (He et al. 2022; Chen et al. 2024), researchers uti-

lize MAEs to handle large amounts of unlabelled graph data. MaskGAE (Li et al. 2023a) selects edges as the masked token while GraphMAE (Hou et al. 2022) focuses on reconstructing the original graph from redundant node features.

Graph Contrastive Learning (GCL) is a well-known representation learning framework that pulls semantically similar instances and pushes semantically different instances. Among them, DGI (Velickovic et al. 2019), GMI (Peng et al. 2020), InfoGCL (Xu et al. 2021a) learn node representations by maximizing the mutual information between the local patch and the global summary of a graph. GRACE (Zhu et al. 2020) and its variants GCA (Zhu et al. 2021), HomoGCL (Li et al. 2023b) adopt SimCLR (Chen et al. 2020) framework for node-level representations. In summary, unsupervised methods, e.g., GAE and GCL provide another possibility to learn from graph data without relying on labeled examples.

Adversarial Attacks and Defense on Graph

Existing studies show that GNNs are vulnerable to adversarial attacks. PGD and min-max methods (Xu et al. 2019) perturb the graph structure from the optimization perspective. *Nettack* (Zügner, Akbarnejad, and Günnemann 2018) is a targeted attack that aims to fool the predictions for specific target nodes. *Metattack* (Zügner and Günnemann 2019) presents a non-targeted attack method based on meta-learning that deteriorates the overall performance of GNNs.

Until now, some mechanisms have been designed to defend against these attacks. Some research focuses on designing novel GNN architectures by adjusting messages from other nodes based on uncertainty (Zhu et al. 2019) or node feature similarity information (Jin et al. 2021). Furthermore, ARIEL (Feng et al. 2022) applies adversarial training to the GCL framework, which trains robust GNN models in an unsupervised manner. SP-AGCL (In, Yoon, and Park 2023) is an adversarial GCL scheme that preserves the node feature similarity. However, as mentioned earlier, these models are exposed to vulnerabilities in situations characterized by node feature attack. Some studies have attempted to handle such attacks under supervised conditions. SG-GSR (In et al. 2024) resists structure-feature attacks by extracting a clean sub-graph. DEGNN (Hasegawa et al. 2024) designs two separate experts to produce modified edges and node features. To sum up, unsupervised GRL methods that can be applied to defend against dual attacks are more challenging but have rarely been studied.

Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V} = (v_1, v_2, ..., v_N)$ and $\mathcal{E}$ represent the sets of nodes and edges respectively. Let $\mathbf{A} \in \{0,1\}^{N \times N}$ represent the adjacency matrix of $\mathcal{G}$, where $\mathbf{A}_{ij} \in \{0,1\}$ indicates the existence of the edge $e_{ij} \in \mathcal{E}$ that links node v_i and v_j. $\mathbf{X} \in \mathbb{R}^{N \times d}$ denotes node feature matrix, and a graph can be denoted as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$. In this work, we assume that $\mathcal{G}$ is under graph structure and feature attacks, and specifically $\mathbf{A}$ is poisoned by adversarial edges and $\mathbf{X}$ is perturbed with noise. The goal of this work is to learn a robust encoder f_E that maps nodes of such poisoned graph into high-quality graph embeddings $\mathbf{H}$ without label supervision, and $\mathbf{H}$ can be used in downstream tasks, e.g., node classification and clustering.

Methodology

As shown in Figure 2, the proposed ACGMAE includes Adversarial-Assisted MAE and Positive-Expansion CL modules.

Adversarial-Assisted Masked Autoencoder Module

Masked Autoencoder The idea of masked autoencoder has been successfully applied for graph self-supervised learning (Hou et al. 2022). Following the paradigm of masked feature reconstruction, we first sample a set of nodes $\widetilde{V} \subset V$ based on Bernoulli distribution and mask each node feature with a mask token [MASK], i.e., a learnable vector $\mathbf{x}_{[M]} \in \mathbb{R}^d$. The masked node feature $\widetilde{\mathbf{X}}$ can be defined as:

$$\widetilde{\mathbf{x}}_i = \begin{cases} \mathbf{x}_{[M]}, & v_i \in \widetilde{V} \\ \mathbf{x}_i, & v_i \in V \end{cases} \tag{1}$$

With an encoder f_E and a decoder f_D, we obtain hidden embeddings $\mathbf{H} = f_E(\mathbf{A}, \widetilde{\mathbf{X}})$ and reconstructed output $\widehat{\mathbf{X}} = f_D(\mathbf{A}, \mathbf{H})$. The node feature reconstruction loss is defined by applying scaled cosine error (SCE) on all the masked nodes:

$$\mathcal{L}_F = \mathcal{L}\left(\mathbf{X}, \widehat{\mathbf{X}}\right) = \frac{1}{|\widetilde{\mathcal{V}}|} \sum_{v_i \in \widetilde{\mathcal{V}}} \left(1 - \frac{\mathbf{x}_i^\top \widehat{\mathbf{x}}_i}{\|\mathbf{x}_i\| \cdot \|\widehat{\mathbf{x}}_i\|}\right)^\gamma \tag{2}$$

where $\gamma \geq 1$ serves as a scaling factor that adjusts convergence speed.

Adversarial Feature Masking Despite its excellent performance, masked autoencoder may be affected by node feature perturbations. We consider adversarial training (In, Yoon, and Park 2023) to mitigate influence of node feature perturbations, and present adversarial feature masking. Specifically, we mask features with small gradients in the negative direction, i.e., change 1s in $\mathbf{X}$ to 0s, which ensures that the original and adversarial features are not far. We first compute the gradient:

$$\mathbf{G}_{\mathbf{X}} = \frac{\partial \mathcal{L}_F}{\partial \widetilde{\mathbf{X}}} = \frac{\partial \mathcal{L}_F}{\partial \mathbf{H}_1} \frac{\partial f_E(\mathbf{A}, \widetilde{\mathbf{X}})}{\partial \widetilde{\mathbf{X}}} \tag{3}$$

Here, $\mathbf{G}_{\mathbf{X}}$ is employed to generate adversarial feature mask $\mathbf{M}$. Specifically, $\mathbf{M}_{ij} = 0$ if the gradient of the j-th feature of the i-th node, i.e., $(\mathbf{G}_{\mathbf{X}})_{ij}$ is small, and $\mathbf{M}_{ij} = 1$ otherwise. The number of zeros in $\mathbf{M}$ is controlled in a perturbation budget, i.e., $\|\mathbf{M}\|_0 \leq \Delta_{\mathbf{X}}$. We generate adversarial node feature matrix $\mathbf{X}_{adv} = \mathbf{M} \odot \widetilde{\mathbf{X}}$, where $\odot$ is hadamard product for matrices, and then define adversarial feature reconstruction loss $\mathcal{L}_{adv} = \mathcal{L}(\mathbf{X}, \widehat{\mathbf{X}}_{adv})$. We regard such adversarial data as attacked node features, and train MAE model to defend against graph feature attack.

Topology Reconstruction In addition to node features, graph also offers complex topological structure. Some studies (Hu et al. 2020; Li et al. 2023a) consider reconstructing graph structure to model connections between nodes effectively. Following such idea, we propose to reconstruct graph topology using the learned embeddings, and enable the reconstructed adjacency matrix $\widehat{\mathbf{A}}$ to be close to $\mathbf{A}$. We have the following graph topology reconstruction loss:

$$\mathcal{L}_E = \frac{1}{N^2}\sum_{i,j}\Big(\widehat{\mathbf{A}}_{ij} - \mathbf{A}_{ij}\Big)^2,\ \widehat{\mathbf{A}} = \text{sigmoid}\left(\mathbf{H}^\top\mathbf{H}\right) \quad (4)$$

Reconstruction loss By combining the above three losses, we have the final reconstruction loss as follows:

$$\mathcal{L}_R = \mathcal{L}_F + \alpha\mathcal{L}_{adv} + \beta\mathcal{L}_E \quad (5)$$

where α and β are two predefined regularization parameters that control the relative importance of the three losses.

Positive-Expansion Contrastive Learning Module

kNN Graph Construction The similarity structure preservation has shown effective to improve the capability of defending against adversarial attacks (Jin et al. 2021). The kNN graph $\mathcal{G}_F = (\mathbf{A}_F, \mathbf{X})$ can be regarded as an augmented view that well preserves similarity structure. Specifically, $\mathbf{A}_F$ is a top-k similarity matrix that is constructed by applying kNN on node feature matrix $\mathbf{X}$ using cosine distance. We then employ f_E to encode $\mathcal{G}_F$ into embeddings, i.e, $\mathbf{Z} = f_E(\mathbf{A}_F, \mathbf{X})$.

Positive-Expansion Contrastive Learning Conventional graph contrastive learning (GCL) (Shen et al. 2023) aims to learn graph representations by maximizing similarity between anchor node and a positive sample, i.e., the corresponding node of anchor node in another view and minimizing similarity between anchor node and negative samples, i.e., all the other nodes except the positive sample. However, the use of single-positive-sample overlooks graph topology, and results in the embeddings that are in conflict with the homophily assumption of GNNs.

To remedy such limitation, we propose to expand positive samples to learn robust representations, and further consider neighbor nodes as positive samples. However, simply assigning neighbor nodes as positive samples is inherently flawed, as adversarial attacks can link unrelated nodes, resulting in the inclusion of false positive samples. To mitigate the effect of adversarial edges, it is expected to estimate the probability of neighbor nodes being true positive (Li et al. 2023b). Specifically, we first apply k-means on $\mathbf{H}$ to assign a cluster label for each node, and construct c centroids $\{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_c\}$. We then calculate the posterior probabilities as follows:

$$p\left(\mathbf{h}_i \mid \mathbf{r}_j\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\|\mathbf{h}_i - \mathbf{r}_j\|_2^2}{2\sigma^2}\right) \quad (6)$$

where σ is standard deviation of Gaussian distribution. By considering an equal prior $p(\mathbf{r}_1) = p(\mathbf{r}_2) = \cdots = p(\mathbf{r}_c)$,

Algorithm 1: Algorithm of ACGMAE

Input: Graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, encoder f_E, decoder f_D.
Parameter: feature perturbation ratio $\Delta_{\mathbf{X}}$, number of nearest neighbors k, number of clusters c, regularization parameters α, β, γ.
Output: f_E, embedding $\mathbf{H}$.

```
Initialize f_E;
Perform knn on X to obtain G_F;
while not converge do
    Calculate embedding H using encoder f_E and reconstructed feature X̂ using decoder ;
    Calculate adversarial node features based on (3) and reconstructed adjacency matrix based on (4);
    Compute reconstruction loss L_R via (2) (4) (5);
    Perform k-means clustering on H and obtain centroids {r_1, r_2, ..., r_c};
    Calculate the soft clustering value for each node-centroid pair according to (7) and get the assignment matrix R;
    Calculate node correlation matrix S via R;
    Compute contrastive loss L_C via (8) (9) (10);
    Compute the final loss L via (12);
    Update the parameters of f_E via L;
end while
return f_E and H = f_E(A, X).
```

the probability of $\mathbf{h}_i$ belonging to the j-th cluster can be calculated via Bayes' rule:

$$p\left(\mathbf{r}_j \mid \mathbf{h}_i\right) = \frac{p\left(\mathbf{r}_j\right)p\left(\mathbf{h}_i \mid \mathbf{r}_j\right)}{\sum_{k=1}^{c} p\left(\mathbf{r}_k\right)p\left(\mathbf{h}_i \mid \mathbf{r}_k\right)} \quad (7)$$

A cluster assignment matrix $\mathbf{R} \in \mathbb{R}^{N\times c}$ is constructed, where $\mathbf{R}_{ij} = p(\mathbf{r}_j \mid \mathbf{h}_i)$ indicates the probability of the i-th node belonging to the j-th cluster. We further calculate a node correlation matrix $\mathbf{S} = \text{norm}\left(\mathbf{R}\right)\cdot\text{norm}\left(\mathbf{R}^\top\right)$ with $\text{norm}\left(\cdot\right)$ being the ℓ_2 normalization on cluster dimension. Therefore, $\mathbf{S}_{ij}$ indicate the connection intensity between the i-th and j-th node, and thus can be estimated as probability of neighbors being true positive.

Based on $\mathbf{S}$, we propose to perform positive-expansion contrastive learning on feature masking graph and kNN graph. Thus global graph structure can be exploited by introducing positive samples. Given $\mathbf{h}_i$ as anchor node, we define the following GCL loss:

$$\ell(\mathbf{h}_i, \mathbf{z}_i) = -\log\frac{\text{pos}}{\text{pos} + \text{neg}} \quad (8)$$

with

$$\text{pos} = e^{\theta(\mathbf{h}_i, \mathbf{z}_i)/\tau} + \sum_{j\in\mathcal{N}_{\mathbf{h}}(i)}\mathbf{S}_{ij}\cdot e^{\theta(\mathbf{h}_i, \mathbf{h}_j)/\tau} \quad (9)$$

$$\text{neg} = \sum_{j\notin\Gamma_{\mathbf{z}}(i)} e^{\theta(\mathbf{h}_i, \mathbf{z}_j)/\tau} + \sum_{j\notin\Gamma_{\mathbf{h}}(i)} e^{\theta(\mathbf{h}_i, \mathbf{h}_j)/\tau} \quad (10)$$

where $\Gamma_{\mathbf{z}}(i) = \{i\cup\mathcal{N}_{\mathbf{z}}(i)\}$ and $\Gamma_{\mathbf{h}}(i) = \{i\cup\mathcal{N}_{\mathbf{h}}(i)\}$, $\mathcal{N}_{\mathbf{h}}(i)$ and $\mathcal{N}_{\mathbf{z}}(i)$ are the neighbor sets of the i-th node in two

Dataset	Setting	GCN	GRACE	DGI	GraphMAE	DGI-Jaccard	STABLE	ARIEL	SP-AGCL	ACGMAE
Cora	Clean	82.19±0.2	81.97±1.2	80.09±0.4	83.26±0.7	81.04+0.9	83.05±0.3	81.77±0.6	84.93±0.9	**84.97±0.6**
	Fea 50%	62.57±1.2	73.36±0.8	60.00±2.5	71.73±0.4	59.06±1.7	75.79±2.4	74.23±1.1	75.69±0.0	**81.01±0.2**
	Meta 25%	66.36±0.5	72.90±0.6	68.87±1.2	72.85±1.1	67.83±0.9	77.43±0.4	76.85±1.8	76.29±0.9	**81.46±0.6**
	Dual Attacks	58.10±0.0	62.30±0.0	59.09±2.3	65.74±0.1	55.84±2.0	48.07±0.5	65.01±3.7	65.16±3.2	**74.55±0.3**
Citeseer	Clean	72.71±0.1	73.96±0.7	73.23±0.7	73.20±0.8	72.88±0.6	74.86±0.3	73.44±1.0	74.96±0.1	**74.97±0.5**
	Fea 50%	66.40±0.8	64.95±0.1	56.67±0.7	61.14±1.2	54.35±5.3	69.52±0.2	52.84±1.8	68.19±1.5	**71.46±0.8**
	Mata 25%	62.90±0.1	64.09±1.3	72.43±0.7	69.57±1.0	72.48±0.7	70.71±0.3	72.54±0.6	**74.84±0.6**	74.74±0.3
	Dual Attacks	60.13±0.1	59.30±0.1	56.50±1.4	59.89±0.1	53.23±1.5	53.47±2.0	52.94±3.0	69.39±0.6	**72.61±0.0**
Pubmed	Clean	**86.31±0.0**	84.12±0.4	85.18±0.3	85.00±0.1	85.04±0.3	84.48±0.1	84.92±0.1	85.41±0.1	85.39±0.2
	Fea 50%	68.49±1.0	66.24±1.0	55.00±0.4	71.29±0.7	55.44±1.3	74.51±0.2	79.08±0.6	67.07±0.7	**80.05±0.2**
	Meta 25%	74.55±0.0	72.64±0.4	75.49±0.4	73.21±0.4	75.15±0.9	75.25±0.3	75.45±0.1	73.06±0.1	**75.50±0.3**
	Dual Attacks	62.57±0.2	61.87±0.0	54.48±0.3	62.23±0.0	55.16±0.3	55.80±0.4	62.87±0.6	51.75±0.3	**65.74±0.3**

Table 1: Node classification performance (Accuracy±Std) under non-targeted attack (*metattack*) and feature attack.

views. The weights ensure the embeddings of two neighbor nodes that have a high likelihood of being true positive samples are close, thereby mitigating the impact of graph structure attack. The final GCL loss is thus defined as:

$$\mathcal{L}_C = \frac{1}{2N}\sum_{i=1}^{N}\left(\ell\left(\mathbf{h}_i, \mathbf{z}_i\right) + \ell\left(\mathbf{z}_i, \mathbf{h}_i\right)\right) \tag{11}$$

Final Objective Function

By combining the above reconstruction and contrastive losses, we have the final loss of ACGMAE as follows:

$$\min \mathcal{L} = \mathcal{L}_R + \gamma \mathcal{L}_C \tag{12}$$

where γ is a predefined regularization parameter to control relative importance of the two losses. The training procedure of ACGMAE is illustrated in Algorithm 1.

Experiments

This section evaluates the performance of the proposed method under graph structure and feature dual attacks. The experiments are performed on a Ubuntu Enterprise 64-Bit Linux workstation with 128G memory and a NVIDIA A6000 GPU server.

Experimental Setup

Datasets We evaluate ACGMAE and baselines on three benchmark datasets, i.e., Cora, Citeseer, Pubmed (Jin et al. 2020), and the largest connected component is considered for each graph.

Baselines We compare the proposed method with several baselines, including four conventional graph representation learning methods, i.e., GCN (Kipf and Welling 2017), GRACE (Zhu et al. 2020), DGI (Velickovic et al. 2019), GraphMAE (Hou et al. 2022), four robust graph representation learning methods, i.e., DGI-Jaccard (Wu et al. 2019), STABLE (Li et al. 2022), ARIEL (Feng et al. 2022), SP-AGCL (In, Yoon, and Park 2023). The implementations of the baselines are kindly provided by the authors.

Attack Methods We use both graph structure and feature attacks to poison graph, and then train model on the poisoned graph to learn graph embeddings. For structure attack, the commonly used *metattack* (Zügner and Günnemann 2019) and *nettack* (Zügner, Akbarnejad, and Günnemann 2018) are employed. For feature attack, following (In et al. 2024), given the i-th node, a noise vector is added to its node feature, where each element is independently sampled from the standard normal distribution.

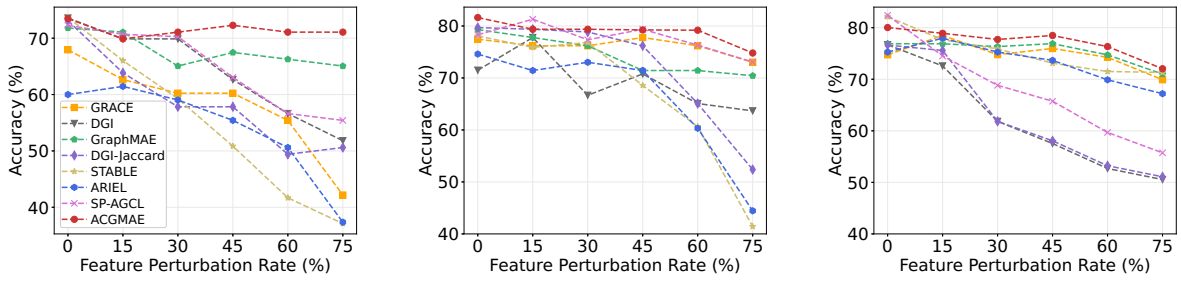
Parameters Setting For all the methods, a two-layer GCN is employed as the encoder, and the default setting is used. In the proposed ACGMAE, the learning rate and weight decay are searched from $\{0.01, 0.001, 0.0001\}$ and $\{0.0001, 0.0005, 0.0001, 0.00005\}$ respectively. The perturbation ratio $\Delta_{\mathbf{X}}$ is searched from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the number of nearest neighbors and the number of clusters are searched from $\{10, 15, 20, 25, 30\}$. The coefficients α, β, and γ are searched from $\{0.01, 0.1, 0.5, 1, 3, 5\}$.

Evaluation Protocol Two downstream tasks, i.e., node classification and node clustering are used to evaluate the quality of learned embeddings. For node classification, we randomly select 10% nodes for training, 10% nodes for validation, and the remaining for testing. We employ logistic regression for node classification, and report classification accuracy. For node clustering, we perform k-means on the learned embeddings, and report Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). The average performance of 5 runs is reported for each experiment.

Node Classification

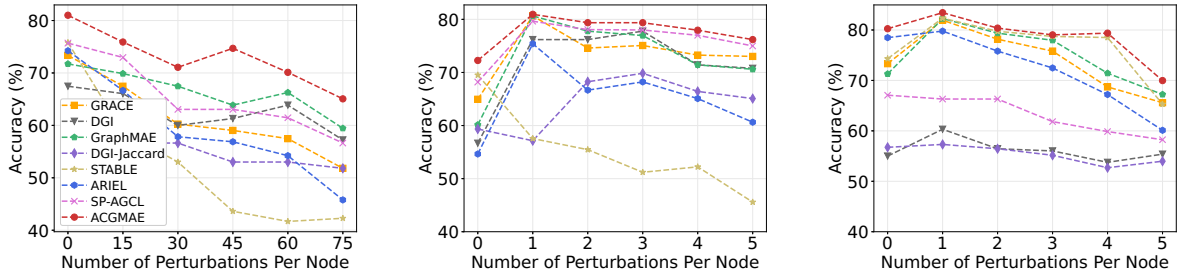
Against fixed attacks The structure and feature perturbation rates are set as 25% and 50% respectively. Table 1 reports the average accuracy with standard deviation of all the methods on three datasets, and the bold indicates the best. We find the following observations:

- The proposed ACGMAE outperforms all the baselines on 10 out of 12 cases. Compared to the baselines, the proposed ACGMAE exhibits robust performance when subjected to incremental structure and feature attacks, benefiting from the advantages of its MAE and CL modules. For instance, under dual attacks, ACGMAE improves the



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

Figure 3: Node classification performance under fixed structure attack (*nettack*) and varying feature attack.



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

Figure 4: Node classification performance under varying structure attack (*nettack*) and fixed feature attack.

best baseline by 8.8%, 2.6%, 2.8% on Cora, Citeseer, Pubmed respectively.

- Existing robust graph representation learning baselines, e.g., STABLE, SP-AGCL defend against structure attack well. However, their performances degrade obviously in feature and dual attacks, as they assume node feature is clean, limiting their robustness and applicability.
- Conventional graph representation learning baselines, e.g., GCN, DGI perform well on clean graphs but perform poorly on attacked graphs, indicating vulnerability of GNNs to graph attacks.

Against varying attacks We empirically compare ACGMAE with the baselines by varying perturbation rates in structure and feature attacks. We choose targeted adversarial attack, i.e., *nettack* as structure attack. We first fix the number of perturbations on every targeted node to 3, vary the feature perturbation rate from 0 to 75% with a step of 15%, and report accuracy in Figure 3. We then fix the feature perturbation rate to 50%, vary the number of perturbations on every targeted node from 0 to 5 with a step of 1, and report average accuracy in Figure 4. From the two figures, we clearly observe that the proposed ACGMAE achieves the best performance in the most cases. As structure and feature perturbation rates increase, the performance drop of ACGMAE is lower than those of the baselines, demonstrating the superior capability of the ACGMAE on defending against structure and feature attacks.

Dataset	Cora		Citeseer	
	NMI	ARI	NMI	ARI
GRACE	18.00	13.11	10.18	16.19
DGI	13.70	14.76	15.25	12.47
GraphMAE	20.80	11.72	11.45	17.82
DGI-Jaccard	15.99	10.53	12.64	10.52
STABLE	32.61	19.99	32.56	26.34
ARIEL	15.71	19.71	14.50	18.91
SP-AGCL	31.27	24.21	22.63	24.89
ACGMAE	**48.10**	**43.59**	**40.99**	**43.19**

Table 2: Node clustering performance (in percentage) under 15% *metattack* and 50% feature attack.

Node Clustering

Table 2 reports node clustering performance of all the methods on the Cora and Citeseer datasets under 15% *metattack* and 50% feature attack. From this table, we observe that the proposed ACGMAE outperforms all the baselines by a large margin in all the cases, revealing its powerful representation capability that benefits clustering. The above empirical results indicate the good performance of the proposed method on more downstream tasks other than node classification.

Ablation Study

This section conducts ablation study of the proposed method by analyzing the effectiveness of adversarial feature

Component			Cora				Citeseer			
AM	TR	PE	Clean	Fea 50%	Mata 25%	Dual Attacks	Clean	Fea 50%	Mata 25%	Dual Attacks
×	×	×	84.23±0.7	71.73±0.4	72.85±1.1	65.74±0.1	73.20±0.8	61.14±1.2	69.57±1.0	59.89±0.1
✓	×	×	84.07±0.9	72.54±0.3	75.77±1.9	67.71±0.1	73.71±0.4	64.16±0.6	72.23±0.2	66.84±0.0
✓	✓	×	84.26±0.9	74.51±0.7	76.21±1.8	68.11±0.1	73.97±0.1	65.87±0.2	72.54±0.3	68.29±0.0
✓	✓	✓	**84.97±0.6**	**81.01±0.2**	**81.46±0.6**	**74.55±0.3**	**74.95±0.6**	**71.46±0.8**	**74.74±0.3**	**72.61±0.0**

Table 3: Ablation study of the proposed method on Cora and Citeseer.

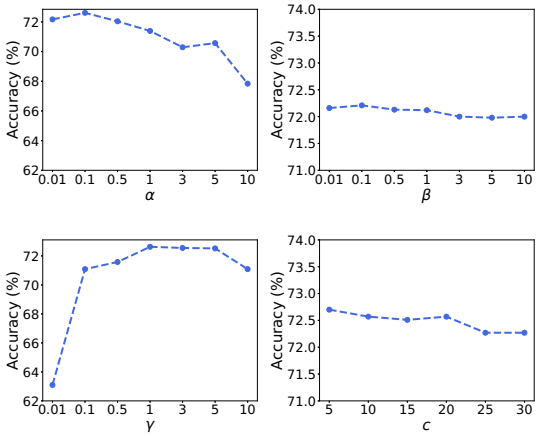
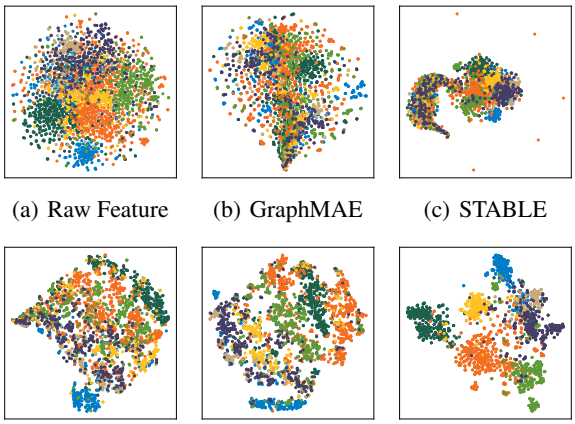


Figure 5: Parameter analysis of the proposed method on Citeseer.



(a) Raw Feature (b) GraphMAE (c) STABLE

(d) ARIEL (e) SP-AGCL (f) ACGMAE

Figure 6: Visualization of the graph embeddings learned by six representative methods on Cora under 25% *metattack* and 50% feature attack.

masking (AM), topology reconstruction (TR) and positive-expansion contrastive learning (PE). We report the performance on Cora and Citeseer in Table 3, where we can clearly observe that adding AM, TR, and PE modules leads to obvious performance improvement, and PE plays the most important roles among the three modules. Specifically, by adding AM module, the performances are improved averagely by 0.2% and 3.1% on clean and attacked graphs respectively; by adding TR module, the performances are improved averagely by 0.2% and 1.0% on clean and attacked graphs respectively; by adding PE module, the performances are improved averagely by 0.8% and 5.1% on clean and attacked graphs respectively. Therefore, the above empirical results clearly demonstrate that AM, TR, and PE modules can help to defend against feature and structure attacks.

Parameter Analysis

This section empirically analyze four important parameters in the proposed method, i.e., regularization parameters α, β, γ, and the number of clusters c. We use *metattack* as structure attack, and set structure and feature perturbation rates to 25% and 50% respectively, and Figure 5 shows accuracy of the proposed method with respect to varying parameters on Cora. As can be observed, the performance is insensitive to the change of β and c. As α increases, the performance decreases a bit, since excessive emphasis on adversarial training can result in inaccurate estimation of the graph. As γ increases, the performance obviously improves, indicating that the CL module with positive sample expansion plays a more significant role to achieve robustness.

Visualization

Figure 6 visualizes graph embeddings learned by six representative methods via t-SNE (Van der Maaten and Hinton 2008) on Cora under 25% *metattack* and 50% feature attack. Each point represents a node and different classes have different colors. As can be observed, compared to the baselines, ACGMAE can generate relatively small clusters, and separate clusters better even under large perturbation. The qualitative empirical results are consistent with previous quantitative empirical results.

Conclusion

In this paper, we have discovered the inability of existing robust graph representation learning methods to resist feature attack. To mitigate this limitation, we propose ACGMAE that effectively defends against graph structure and feature dual attacks. By introducing adversarial feature masking in MAE and designing a positive-expansion strategy in CL, the robustness of the model has been improved significantly. Extensive empirical studies on various graph structure and feature dual attacks verify the effectiveness of the proposed ACGMAE. As future work, we aim to develop robust graph representation method to defend against other graph attacks, e.g., label attack.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62472226, 62176126, the Natural Science Foundation of Jiangsu Province, China under Grant No. BK20230095.

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; and Wang, J. 2024. Context autoencoder for self-supervised representation learning. *IJCV*, 132(1): 208–223.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial attack on graph structured data. In *ICML*, 1115–1124.

Feng, S.; Jing, B.; Zhu, Y.; and Tong, H. 2022. Adversarial Graph Contrastive Learning with Information Regularization. In *WWW*, 1362–1371.

Geisler, S.; Zügner, D.; and Günnemann, S. 2020. Reliable Graph Neural Networks via Robust Aggregation. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NIPS*, volume 33, 13272–13284.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*, 1263–1272.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-mixup: Graph data augmentation for graph classification. In *ICML*, 8230–8248.

Hasegawa, T.; Yun, S.; Liu, X.; Phua, Y. J.; and Murata, T. 2024. DEGNN: Dual Experts Graph Neural Network Handling both Edge and Node Feature Noise. In *PAKDD*, 376–389.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, 16000–16009.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *SIGKDD*, 594–604.

Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; and Sun, Y. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *SIGKDD*, 1857–1867.

In, Y.; Yoon, K.; Kim, K.; Shin, K.; and Park, C. 2024. Self-Guided Robust Graph Structure Refinement. In *WWW*, 697–708.

In, Y.; Yoon, K.; and Park, C. 2023. Similarity preserving adversarial graph contrastive learning. In *SIGKDD*, 867–878.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node similarity preserving graph convolutional networks. In *WSDM*, 148–156.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *SIGKDD*, 66–74.

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. arXiv:1611.07308.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, J.; Wu, R.; Sun, W.; Chen, L.; Tian, S.; Zhu, L.; Meng, C.; Zheng, Z.; and Wang, W. 2023a. What's Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In *SIGKDD*, 1268–1279.

Li, K.; Liu, Y.; Ao, X.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Reliable representations make a stronger defender: Unsupervised structure refinement for robust gnn. In *SIGKDD*, 925–935.

Li, W.-Z.; Wang, C.-D.; Xiong, H.; and Lai, J.-H. 2023b. Homogcl: Rethinking homophily in graph contrastive learning. In *SIGKDD*, 1341–1352.

Ma, T.; Chen, J.; and Xiao, C. 2018. Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *NIPS*, volume 31.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*, 259–270.

Shen, X.; Sun, D.; Pan, S.; Zhou, X.; and Yang, L. T. 2023. Neighbor contrastive learning on learnable graph augmentation. In *AAAI*, volume 37, 9782–9791.

Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Philip, S. Y.; He, L.; and Li, B. 2022. Adversarial attack and defense on graph data: A survey. *IEEE TKDE*, 35(8): 7693–7711.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*, 9(11).

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. In *ICLR*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep graph infomax. In *ICLR*.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019. Adversarial examples for graph data: deep insights into attack and defense. In *IJCAI*, 4816–4823.

Xu, D.; Cheng, W.; Luo, D.; Chen, H.; and Zhang, X. 2021a. Infogcl: Information-aware graph contrastive learning. *NIPS*, 34: 30414–30425.

Xu, H.; Xiang, L.; Yu, J.; Cao, A.; and Wang, X. 2021b. Speedup Robust Graph Structure Learning with Low-Rank Information. In *CIKM*, 2241–2250.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. In *IJCAI*, 3961–3967.

Zhang, M.; and Chen, Y. 2018. Link Prediction Based on Graph Neural Networks. In *NIPS*, volume 31.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *SIGKDD*, 1399–1407.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep Graph Contrastive Representation Learning. arXiv:2006.04131.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*, 2069–2080.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *SIGKDD*, 2847–2856.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. arXiv:1902.08412.