Postprint of Adversarial Training Method for Graph Convolutional Neural Networks Based on Non-Robust Features

Authors: Chengqi, Zhu Hongliang, Xin Yang

Date: 2022-04-07T15:01:57Z

Abstract

Graph Convolutional Networks (GCNs) can extract effective information from graph data through graph convolutions, but they are vulnerable to adversarial attacks that degrade model performance. While adversarial training can enhance neural network robustness, the discrete nature of graph structures and node features precludes direct gradient-based construction of adversarial perturbations. Extracting graph data features from the model' s embedding space as adversarial training samples can reduce this construction complexity. Inspired by ensemble learning, we propose VDERG, an adversarial training method for GCNs based on non-robust features. Specifically, we construct two GCN submodels targeting topological structure and node attributes, respectively. Non-robust features are extracted through the embedding space to perform adversarial training, and the embedding vectors output by the two sub-models are finally integrated as node representations. Experimental results demonstrate that the proposed method improves accuracy by an average of 0.8% on clean data and by up to 6.91% under adversarial attacks.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers
ChinaXiv Cooperative Journal

Graph Neural Networks Adversarial Training with Non-Robust Features

Cheng Qi, Zhu Hongliang†, Xin Yang
(School of Cyber Security, Beijing University of Posts and Telecommunications,

Beijing 100876, China)

Abstract: Graph convolutional neural networks can distill effective information from graph data through graph convolution, but they are vulnerable to adversarial attacks that degrade model performance. Adversarial training can improve neural network robustness, yet since graph structures and node features are typically discrete, adversarial perturbations cannot be directly constructed based on gradients. Extracting graph data features from the model' s embedding space as adversarial training samples can reduce construction complexity. Drawing on ensemble learning principles, this paper proposes VDERG, an adversarial training method for graph convolutional neural networks based on non-robust features. VDERG constructs two graph convolutional neural network sub-models targeting topology structure and node attributes respectively, extracts non-robust features through the embedding space, completes adversarial training based on these features, and finally integrates the embedding vectors output by the two sub-models as node representations. Experimental results demonstrate that the proposed adversarial training method improves accuracy on clean data by 0.8% on average and enhances accuracy by up to 6.91% under adversarial attacks.

Keywords: graph convolutional neural network; ensemble learning; non-robust features; adversarial training

0 Introduction

Graphs, as a universal data structure, can widely represent systems across various domains, including economic networks (transaction networks), social sciences (social and citation networks), natural sciences (molecular structures), and knowledge graphs. In recent years, graph neural networks (GNNs) have achieved remarkable success in learning graph representations. Among them, graph convolutional neural networks (GCNs) generate node representations by aggregating node information using edge information, demonstrating significant effectiveness in graph information extraction. The extracted features can be applied to node classification, link prediction, graph classification, and other tasks, with broad applications in data mining and recommendation systems.

Existing research has demonstrated that neural networks lacking robustness are susceptible to adversarial attacks, where adversarial samples with minimal perturbations substantially degrade neural network performance [**?**]. Dai et al. [**?**] discovered that randomly dropping edges between nodes can effectively attack graph neural networks. The vulnerability of GCNs may lead to security issues in their application domains; for instance, in credit detection systems, fraudsters could establish multiple transactions with several high-credit users to obtain false "high-credit user" results in model detection [**?**]. Consequently, extensive research has emerged focused on improving GCN robustness.

Adversarial training [**?**] has been widely employed to enhance neural network robustness by generating adversarial samples during model training and minimizing model loss on these samples, thereby improving performance under adversarial attacks. Existing research on adversarial training methods for GCNs primarily concentrates on constructing perturbation regularization terms for single models or modifying graph structures, with few studies exploring ensemble-based approaches to improve model robustness by leveraging the learning capabilities of multiple classifiers.

A key characteristic of adversarial attacks is their generalizability across neural networks. By performing adversarial training on multiple neural network models separately, the overall model can learn more comprehensive feature information, thereby enhancing robustness. Literature [**?**, **?**] indicates that the effectiveness of ensemble learning-based defense algorithms depends on the diversity of sub-models. Only when sub-models learn different features can adversarial perturbations be prevented from transferring between sub-models, effectively improving the overall model' s defense capability. Considering graph data characteristics, Wu et al. [**?**] constructed an ensemble model containing two sub-models trained separately on topology structure information and node attribute information to enhance GNN robustness. However, merely training sub-models separately on structural and attribute information without considering the characteristics of adversarial attacks may still produce significant prediction deviations when both structural and attribute information are attacked.

Model training on neural networks essentially involves learning features from graph data. The learned features that benefit model performance exhibit varying sensitivity to adversarial perturbations. Based on this differential sensitivity, these features can be categorized into robust and non-robust features. Robust features in data maintain stability even under adversarial attacks, helping models learn correct and effective information, whereas non-robust features are altered by adversarial perturbations, causing models to learn incorrect information during training and consequently degrading performance. The non-robust features learned by models contribute to their vulnerability. However, current research on adversarial training based on non-robust features has focused on image data, with few studies leveraging non-robust features in graph data to improve model robustness.

Addressing the aforementioned issues, this paper aims to explore the role of non-robust features in graph data for improving GCN model robustness. Combining structural information and node attribute information in graph data, we provide definitions and extraction methods for non-robust features. Furthermore, based on non-robust features learned by GCNs from graph data and ensemble learning methods, we propose VDERG (Vulnerabilities Distillation of Ensembles for Robust Graph Neural Networks). VDERG utilizes embedding vectors after graph convolutional layers to extract non-robust features from structural and attribute information separately, performs adversarial training on two sub-models based on these features to make them respectively adapt to adversarial

perturbations on node relationships and node attributes, and then integrates the node embedding vectors from both sub-models as input to a mapping function for final predictions. Experiments demonstrate that the proposed defense algorithm can effectively improve the robustness of graph convolutional neural network models.

The main contributions of this paper are: (a) defining non-robust features on graph data and providing extraction methods for non-robust features from both structural and attribute perspectives considering graph data characteristics; (b) proposing a robust graph convolutional neural network algorithm based on ensemble learning that performs adversarial training on sub-models using non-robust features, enables different sub-models to learn graph information from structural and attribute information respectively, integrates node vector representations, and effectively defends against adversarial attacks.

1.1 Adversarial Training

Graph convolutional neural networks can be viewed as a variant of convolutional neural networks migrated to graph data. Due to their similar convolution mechanisms, GCNs are also vulnerable to adversarial attacks. For a graph $\mathcal{G}$, an attacker aims to find a graph structure $\mathcal{G}'$ that maximizes the loss value $\mathcal{L}$ for target node v on GCN model f, i.e., $\mathcal{G}' = \arg\max_{\mathcal{G}'} \mathcal{L}(f(\mathcal{G}'), y_v)$. The adversarial perturbation must be constrained to be imperceptible, i.e., $\|\mathcal{G}' - \mathcal{G}\| \leq \epsilon$. Due to the characteristics of graph data tasks, most current adversarial attacks are poisoning attacks, where attackers inject adversarial samples into the training dataset [**?**].

In recent years, with the powerful expressive capability of GCNs in node representation, numerous studies have focused on improving GCN model robustness. Adversarial training has achieved significant success in improving the robustness of CNNs and other models, and has been adapted by many scholars to enhance GCN robustness. Adversarial training generates adversarial samples during model training and simultaneously minimizes model loss on these samples, i.e., $\min_\theta \mathbb{E}_{(\mathcal{G},y)}[\max_{\mathcal{G}':\|\mathcal{G}'-\mathcal{G}\|\leq\epsilon} \mathcal{L}(f_\theta(\mathcal{G}'), y)]$.

Dai et al. [**?**] perturbed the adjacency matrix by randomly dropping edges during training, but this method only reduced the attack success rate by 1%. Dai et al. [**?**] proposed adversarial training for poisoning attack scenarios by adding noise in the embedding space based on DeepWalk [**?**], improving DeepWalk' s generalization ability on node classification tasks. This adversarial training method can be extended to a series of node embedding models, but the experiments lacked comparative verification of model robustness. Feng et al. [**?**] argued that graph smoothness causes adversarial perturbations to propagate between nodes, and addressed this issue by adding an adversarial regularization term to reduce the difference between target samples and their adjacent samples' predictions. Results showed that GCN-GAD with the added regularization term

became less sensitive to adversarial perturbations, but the experiments did not clearly specify the adversarial attack methods used.

Wang et al. [**?**] proposed ignoring graph discreteness and directly adding perturbations to the adjacency matrix and feature matrix. Targeting a random edge-dropping attack method, experiments verified that the proposed GraphDefense method could improve model accuracy by approximately 0.2 after adversarial attacks. The discreteness of graph data poses challenges for adversarial training on GCNs, and directly adding perturbations to the adjacency or feature matrix can reduce the complexity of adversarial training methods.

1.2 Ensemble Learning

Ensemble learning has been widely studied for improving model performance by combining multiple base learners to enhance overall model generalization. Since neural network models tend to extract similar features from datasets, adversarial attacks also exhibit generalizability across different graph neural networks [**?**]. Ensemble learning-based defense methods can prevent the impact of adversarial attacks from transferring between sub-models by making different sub-models have different adversarial subspaces (Adv-SS) [**?**]. Kariyappa et al. [**?**] proposed diversity training to reduce the correlation of loss functions between sub-models. Pang et al. [**?**] proposed an adaptive regularization term that encourages diversity in the non-maximal predictions of different sub-models. Yang et al. [**?**] discovered that non-robust features are more widely distributed in data and improved model performance on both clean and attacked data by having sub-models extract different non-robust features and integrating model learning capabilities. The aforementioned ensemble learning methods have achieved significant results in the image domain.

Currently, few studies have applied ensemble learning to the graph domain to improve model performance and robustness. Zhang et al. [**?**] reconstructed an attribute graph based on feature similarity between nodes and performed predictions based on structural information and the attribute graph separately, finally aggregating the two predictions as the result. This ensemble algorithm is based on the assumption that nodes with similar features and adjacent nodes usually have similar labels, preprocesses attribute information to some extent, and improves model performance, but cannot eliminate attack effects when graph structure is perturbed, exhibiting certain limitations. Wu et al. [**?**] selected two sub-models to learn from graph structural information and attribute information respectively, averaged the confidence of the two sub-models in each iteration, used the ensemble model' s most confident prediction as the node' s pseudo-label, and added it to the training set to improve model robustness. This method primarily addresses the lack of labels in semi-supervised learning and does not consider changes in graph structure and node attributes under adversarial attacks.

1.3 Non-Robust Features Research

Under supervised learning, neural networks improve model capability by extracting and learning features from datasets. The features learned by neural networks directly determine their predictive ability. Ilyas et al. [**?**] argued that well-generalized features learned by models form the basis of adversarial attacks. By constructing "robust datasets" and "non-robust datasets" on image data, they demonstrated that non-robust features in datasets cause neural networks to be vulnerable to adversarial attacks, making non-robust features valuable for research in improving neural network robustness. Yang et al. [**?**] extracted non-robust features from image data through embedding vectors after model convolutional layers, improving model robustness while maintaining performance on clean datasets.

Current research on non-robust features has primarily focused on the image domain. However, Garg et al. [**?**] discovered that robust features unaffected by adversarial attacks are related to the spectral characteristics of image data, suggesting that the Laplacian matrix of graph data may also contain non-robust features that contribute to GCN vulnerability. Jin et al. [**?**] experimentally demonstrated that removing adversarial attack edges and normal edges has different effects on the rank and singular values of the adjacency matrix, indicating that the features exploited in adversarial attack generation have particular characteristics, indirectly confirming that features learned by GCNs have varying susceptibility to adversarial attacks. Since GCNs perform model training based on both structural information and node attribute information, research on non-robust features in graph data should address these two aspects. Literature [**?**] compared real-world graphs with graphs attacked by metattack [**?**] and found that adjacent nodes in real graphs tend to have similar attribute features, while adversarial attacks alter graph smoothness. Literature [**?**] improved model performance by constructing regularization terms that enhance graph smoothness. The aforementioned research suggests that non-robust features in graph datasets may be related to graph smoothness.

2 Ensemble Adversarial Training Method Based on Non-Robust Features

Inspired by non-robust feature extraction methods in the image domain, this paper proposes VDERG (Vulnerabilities Distillation of Ensembles for Robust Graph Neural Networks), an ensemble adversarial training method based on non-robust features. Considering both topology structure and node attribute information in graph data, VDERG obtains gradients in the model' s embedding vector space through matrix differences with random graphs and feature

smoothness differences respectively, performing iterations on the adjacency matrix and attribute matrix to extract non-robust features from graph data. These non-robust features serve as adversarial samples for adversarial training of two sub-models, which learn from structural non-robust features and attribute non-robust features respectively. Finally, VDERG sums and averages the embedding vectors from the two sub-models, obtaining node prediction labels through a softmax function. The overall process is illustrated in Figure 1.

2.1 Problem Formulation

A graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set containing N nodes, and $\mathcal{E}$ is the edge set. Node relationships can be represented by adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where A_{ij} represents the relationship between node v_i and node v_j. $\mathbf{X} \in \mathbb{R}^{N \times d}$ denotes the node feature matrix, where $\mathbf{x}_i$ represents the feature vector of node v_i. According to common node classification task settings, this paper assumes that only partial node labels are available in the dataset, denoted as $\mathcal{Y}_L$, where node v_i' s label corresponds to y_i. For node classification tasks, given graph $\mathcal{G}$ and partial node labels $\mathcal{Y}_L$, GCN aims to learn a function f_θ that maps nodes to a set of labels, using the function to classify unlabeled nodes. The learning process can be described by:

$$\mathcal{L} = \sum_{v_i \in \mathcal{Y}_L} \ell(f_\theta(\mathbf{A}, \mathbf{X})_i, y_i)$$

where $f_\theta(\mathbf{A}, \mathbf{X})_i$ represents the prediction for node v_i, θ denotes the learnable parameters, and ℓ represents the difference between predictions and labels, typically calculated using cross-entropy. The most commonly used GCN structure is a two-layer GCN [**?**], i.e., model parameters $\theta = \{\mathbf{W}_1, \mathbf{W}_2\}$, so function f_θ can be further specified as:

$$f_\theta(\mathbf{A}, \mathbf{X}) = \text{softmax}(\hat{\mathbf{A}} \cdot \text{ReLU}(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}_1) \cdot \mathbf{W}_2)$$

where $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ represents the normalized adjacency matrix, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$; $\tilde{\mathbf{D}}$ is the diagonal degree matrix; σ denotes the activation function, commonly ReLU.

Based on the above definitions, given graph $\mathcal{G}$ and labels $\mathcal{Y}_L$, the VDERG algorithm proposed in this paper targets poisoning attacks. Under the premise that adjacency matrix $\mathbf{A}$ and feature matrix $\mathbf{X}$ may be poisoned, VDERG learns GCN model parameters θ through adversarial training to obtain a robust GCN model that improves prediction performance on unlabeled nodes under adversarial attacks.

2.2 Non-Robust Feature Extraction

GCNs learn node embedding representations by extracting features from graph data. Among the graph data features utilized during extraction, some features are robust—meaning they are not easily affected by adversarial perturbations —while others are non-robust, causing model performance degradation when attacked.

Consider the ideal scenario for non-robust feature extraction: the extracted perturbed graph contains all possible non-robust features that could interfere with GCN, and although the difference between the original graph data and perturbed graph data is substantial, they produce identical embedding vectors after passing through the GCN model, as shown in Figure 2. In this case, the non-robust features contained in the perturbed graph would have a fatal impact on GCN' s node embedding generation. Based on this theory, this paper defines non-robust features extracted by GCN from graph data as follows:

Let $\mathcal{G}$ be the original graph data' s adjacency and attribute matrices, and $\mathcal{G}'$ be a randomly generated graph with the same number of nodes but different node relationships and attribute features. The non-robust features F_{NR} extracted by GCN model f_θ at layer l from graph $\mathcal{G}$, corresponding to graph $\mathcal{G}'$, can be defined as:

$$F_{NR} = \arg\min_{\mathcal{G}'} \|f_\theta^l(\mathbf{A}', \mathbf{X}') - f_\theta^l(\mathbf{A}, \mathbf{X})\|_F^2 + \lambda \cdot \text{Smoothness}(\mathbf{X}', \mathbf{X})$$

where $f_\theta^l(\cdot)$ represents the output before the activation function (e.g., ReLU) at the l-th hidden layer of the GCN model. Considering that adversarial attacks can interfere with GCN models by modifying node relationships or node attributes, the feature extraction process in Equation (3) performs constrained optimization from both adjacency matrix and attribute matrix perspectives, aiming to extract from graph $\mathcal{G}'$ features that could confuse GCN into identifying it as graph $\mathcal{G}$, i.e., non-robust features in graph $\mathcal{G}$.

The first term in Equation (3) minimizes the difference between the original graph' s adjacency matrix and extracted adjacency features in the embedding space, making the features extracted from node relationships approximate the node relationship information learned by GCN. This objective can be achieved by minimizing the Frobenius norm of the difference between $f_\theta^l(\mathbf{A}', \mathbf{X})$ and $f_\theta^l(\mathbf{A}, \mathbf{X})$, i.e., the first term can be rewritten as $\|f_\theta^l(\mathbf{A}', \mathbf{X}) - f_\theta^l(\mathbf{A}, \mathbf{X})\|_F^2$.

The second term considers extracting features from node attribute information. Adversarial attacks reduce graph smoothness when connecting nodes with large attribute differences or deleting links between similar nodes. Therefore, this paper minimizes the feature smoothness difference between the original graph attribute matrix $\mathbf{X}$ and extracted attribute features $\mathbf{X}'$, making the features extracted from node attributes approximate the node attribute information learned by GCN. The second term in Equation (3) can be rewritten

as $\|\mathbf{X}'^T\mathbf{L}\mathbf{X}' - \mathbf{X}^T\mathbf{L}\mathbf{X}\|_F^2$, where $\mathbf{L}$ is the graph Laplacian matrix. Similarly, Smoothness$(\mathbf{X}', \mathbf{X})$ measures the feature smoothness difference between nodes. By constraining feature smoothness differences for attribute feature extraction, this method fully considers the attack characteristic that adversaries often connect dissimilar nodes to reduce model prediction capability.

2.3 Ensemble Adversarial Training Based on Non-Robust Features

Ensemble learning can improve model robustness as a training strategy. By having sub-models in an ensemble learn different features, model performance can be enhanced while maintaining simple model structures. If different sub-models can learn different non-robust features, the generalizability of adversarial attacks can be prevented from affecting all sub-models, improving the performance of the integrated model. Based on this theory, this paper adopts the ensemble learning concept, using two sub-models to extract non-robust features from graph data from node relationship and node attribute perspectives respectively, and performs adversarial training using the extracted non-robust features.

Adversarial training typically adds small perturbations to samples to make neural networks adapt to perturbations and improve robustness on adversarial samples. However, as graph data is non-Euclidean, adversarial samples cannot be constructed through gradient-based methods. Therefore, performing adversarial training through extracted features avoids the data discreteness issues in adversarial sample construction, making the approach simpler and more interpretable.

2.3.1 Non-Robust Feature Learning Method Based on Node Relationships

Referring to Equations (1) and (2), the process of the first sub-model extracting non-robust features contained in the adjacency matrix can be expressed as:

$$\mathcal{L}_{\text{struct}} = \|\mathbf{A}' - \mathbf{A}\|_F^2 + \lambda_1 \|f_{\theta_1}^2(\mathbf{A}', \mathbf{X}) - f_{\theta_1}^2(\mathbf{A}, \mathbf{X})\|_F^2$$

where $f_{\theta_1}^2(\cdot)$ represents the embedding vector after the second convolutional layer and before the activation function of the first sub-model. By constraining the difference between feature $\mathbf{A}'$ and random graph adjacency matrix $\mathbf{A}'$ to be less than ϵ, and minimizing the distance difference between $\mathbf{A}'$ and original graph adjacency matrix $\mathbf{A}$ in the embedding space, non-robust features are extracted from random graph $\mathcal{G}''$ s adjacency matrix that are similar to $\mathcal{G}'$ but would mislead the GCN model into predicting $\mathcal{G}$. The target loss function for adversarial training of the first sub-model is:

$$\mathcal{L}_1 = \mathcal{L}_{\text{CE}}(f_{\theta_1}(\mathbf{A}', \mathbf{X}), \mathbf{Y}) + \lambda_2 \mathcal{L}_{\text{struct}}$$

where $\mathcal{L}_{\text{CE}}(f_{\theta_1}(\mathbf{A}', \mathbf{X}), \mathbf{Y})$ is the loss function of the first GCN sub-model on input features $(\mathbf{A}', \mathbf{X})$. Minimizing Equation (7) trains the first model to learn non-robust features contained in node relationships, improving model robustness.

2.3.2 Non-Robust Feature Learning Method Based on Node Attributes

Referring to Equations (1) and (3), the second sub-model similarly extracts non-robust features contained in the attribute matrix based on embedding vectors after the second convolutional layer. This process can be expressed as:

$$\mathcal{L}_{\text{attr}} = \|\mathbf{X}' - \mathbf{X}\|_F^2 + \lambda_3 \|\mathbf{X}'^T \mathbf{L} \mathbf{X}' - \mathbf{X}^T \mathbf{L} \mathbf{X}\|_F^2$$

where $f_{\theta_2}^2(\cdot)$ represents the embedding vector after the second convolutional layer and before the activation function of the second sub-model. Similarly, by constraining the difference between feature $\mathbf{X}'$ and random graph attribute matrix $\mathbf{X}'$ to be less than ϵ, and minimizing the feature smoothness difference between $\mathbf{X}'$ and original graph attribute matrix $\mathbf{X}$ in the embedding space, non-robust features corresponding to graph $\mathcal{G}$ are extracted from random graph $\mathcal{G}'$' s attribute matrix. The target loss function for adversarial training of the second sub-model is:

$$\mathcal{L}_2 = \mathcal{L}_{\text{CE}}(f_{\theta_2}(\mathbf{A}, \mathbf{X}'), \mathbf{Y}) + \lambda_4 \mathcal{L}_{\text{attr}}$$

where $\mathcal{L}_{\text{CE}}(f_{\theta_2}(\mathbf{A}, \mathbf{X}'), \mathbf{Y})$ is the loss function of the second GCN sub-model on input features $(\mathbf{A}, \mathbf{X}')$. Equation (9) trains the second sub-model to learn non-robust features from the node attribute perspective, reducing the impact of adversarial attacks.

2.3.3 Ensemble Learning-Based Adversarial Training Strategy

Based on the above non-robust feature learning method, the adversarial training process of VDERG proposed in this paper is as follows: First, randomly initialize two GCN sub-models. In each iteration, generate random graphs with the same number of nodes as the input graph, and use stochastic gradient descent to optimize Equations (6) and (8) to extract non-robust features of the input graph from the adjacency matrix and attribute matrix respectively using random graphs. Then perform adversarial training on the two sub-models separately based on non-robust features in node relationships and node attributes, optimize sub-model parameters using the cross-entropy loss functions in Equations (7) and (9), and optimize network parameters using Adam. Finally, sum

and average the node embedding vectors obtained from the two sub-models, and obtain the final model prediction results through a softmax function. The pseudocode is shown in Algorithm 1.

Algorithm 1: VDERG Training Strategy

Input: Adjacency matrix $\mathbf{A}$, attribute matrix $\mathbf{X}$, labels $\mathbf{Y}$, feature extraction iteration counts T_1, T_2, step sizes α_1, α_2, learning rate η.

Output: Integrated GCN model parameters θ_1, θ_2, node prediction results.

1. Randomly initialize θ_1, θ_2 // Initialize parameters for 2 GCN sub-models
2. for epoch in training_epochs:
3. `/* Generate random graph by applying random attack with perturbation rate 1.0 on input`
4. `$\mathcal{G}' \leftarrow \text{RandomAttack}(\mathcal{G})$`
5. `Initialize $\mathbf{A}' \leftarrow \mathbf{A}_{\text{rand}}$ // Initialize features usi`
6. `for $t_1$ in $T_1$:`
7. `    Update $\mathbf{A}'$ using Equation (6) with step size $\alpha_1$ // Extract non-ro`
8. `$\mathbf{H}_1 \leftarrow f_{\theta_1}^2(\mathbf{A}', \mathbf{X})$ // Obtain embedding v`
9. `Update $\theta_1$ using Equation (7) // Update first sub-model parameters`
10. Initialize $\mathbf{X}' \leftarrow \mathbf{X}_{\text{rand}}$ // Initialize features using random graph's attribute matrix
11. for t_2 in T_2:
12. `    Update $\mathbf{X}'$ using Equation (8) with step size $\alpha_2$ /* Extract non-rob`
13. $\mathbf{H}_2 \leftarrow f_{\theta_2}^2(\mathbf{A}, \mathbf{X}')$ // Obtain embedding vector based on attribute non-robust features
14. Update θ_2 using Equation (9) // Update second sub-model parameters
15. $\mathbf{H}_{\text{ensemble}} \leftarrow (\mathbf{H}_1 + \mathbf{H}_2)/2$
16. $\hat{\mathbf{Y}} \leftarrow \text{softmax}(\mathbf{H}_{\text{ensemble}})$

3.1 Dataset Description

This paper selects three common citation network datasets in the graph domain for node classification task experiments. Dataset details are shown in Table 1.

Table 1: Dataset Description

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3

In experiments, following the dataset splitting method of the renowned attack algorithm Metattack, this paper randomly splits all datasets into labeled and unlabeled sets with a 10% and 90% ratio, then further divides the labeled set into training and validation sets with a 50% and 50% ratio.

3.2 Model Performance Comparison

To verify the adversarial attack defense capability of the proposed VDERG, this paper evaluates VDERG against several state-of-the-art GCN defense algorithms in terms of node classification accuracy based on the Metattack adversarial attack algorithm. Metattack has five variants. On the Cora and Citeseer datasets, this paper uses the Meta-Self variant with the best attack effect for experiments; on the PubMed dataset, to save time and memory, this paper uses the A-Meta-Self variant similar to Meta-Self. Experiments are conducted for perturbation rates from 0 to 20%, increasing by 5% each time. Reference experimental results are shown in Table 2, where results for GCN, GAT, GCN-Jaccard, and Pro-GNN are from literature [**?**], and SimP-GCN results are from the original paper. To make model results more objective and eliminate randomness in deep learning training, all experiments are repeated 10 times.

Table 2: Node Classification Performance Comparison Under Global Attack (Metattack)

Perturbation Rate (%)	Cora	Citeseer	PubMed
0	83.50±0.44	76.55±0.79	87.19±0.09
5	70.39±1.28	65.10±0.71	83.09±0.13
10	59.56±2.72	64.52±1.11	81.21±0.09
15	-	62.03±3.49	78.66±0.12
20	-	-	77.35±0.19

Note: The table shows comparative results. VDERG achieves 84.26±0.43, 75.01±1.09, and 87.91±0.23 on clean data (0% perturbation) for Cora, Citeseer, and PubMed respectively.

The results in Table 2 show that at 0% perturbation rate, VDERG improves model accuracy on the Cora, Citeseer, and PubMed datasets by 0.84%, 1.25%, and 0.32% respectively compared to current best models, demonstrating that

VDERG can more comprehensively learn graph data information by integrating node attribute and structural features, and that adversarial training with non-robust features not only improves model robustness but also enhances performance on clean datasets.

For perturbation rates from 5% to 20%, VDERG achieves higher accuracy than existing best models on all three datasets. The increase in perturbation rate does not cause VDERG' s accuracy to drop as significantly as the original GCN. During the perturbation rate increase, VDERG' s model accuracy declines more slowly compared to other methods, showing stronger robustness. Compared to other classifiers, VDERG' s performance improvement is most pronounced on the Cora dataset. At a 20% perturbation rate, VDERG' s accuracy is 6.91% higher than the current best model.

3.3 Comparison of Ensemble and Single Feature Learning

To study the effectiveness of the ensemble method in improving model performance, this subsection compares model performance when considering only structural information or only attribute information during the ensemble process. Experimental results are shown in Table 3, which presents results on Cora and Citeseer datasets for VDERG-structure (extracting non-robust features only from structural information) and VDERG-features (extracting non-robust features only from attribute information).

Table 3: Comparison of Structure and Features Ablation

Dataset	Perturbation Rate (%)	VDERG-structure	VDERG-features	VDERG
Cora	0	85.51±0.30	74.46±1.06	84.26±0.43
	5	83.82±0.75	73.31±1.04	83.98±0.63
	10	82.10±0.85	73.02±0.62	82.72±1.38
	15	81.69±1.36	73.34±0.47	81.70±0.71
	20	80.19±1.31	72.18±0.80	80.23±1.21
Citeseer	0	84.48±0.53	73.69±1.84	75.01±1.09
	5	84.26±0.43	73.02±0.50	74.16±0.66
	10	83.65±1.21	72.71±1.57	73.76±0.38
	15	83.98±0.63	72.86±0.95	73.52±0.81
	20	81.50±1.40	-	73.41±1.23

The table shows that although VDERG' s performance on clean Cora data is slightly inferior to considering structural information alone, VDERG achieves the best classification results on the Citeseer dataset and under adversarial attacks, demonstrating that the proposed ensemble strategy can effectively improve model robustness and graph information representation capability under

adversarial attacks. Additionally, experimental results indicate that adversarial training based solely on non-robust features from structural information performs better than methods based only on attribute information. This is because extracting non-robust features based on feature smoothness differences may cause over-smoothing of isolated nodes, and VDERG' s comprehensive consideration of structural information can effectively compensate for this deficiency.

3.4 Comparison of Different Model Parameters

For the proposed VDERG strategy, the efficiency of non-robust feature extraction is crucial. Therefore, this subsection analyzes the impact of step size α_1 and iteration count T_1 in structural non-robust feature extraction, and step size α_2 and iteration count T_2 in attribute non-robust feature extraction on VDERG performance. Experiments are conducted on the Cora dataset under Metattack with 10% perturbation rate. Results are shown in Figure 3. The variation range for step sizes α_1 and α_2 is set from 5e-5 to 1, and iteration counts T_1 and T_2 from 1 to 12.

Figure 3 shows that in both non-robust feature extraction processes, model performance first increases and then decreases with iteration count. For structural information feature extraction, the optimal iteration count is 7. For attribute information feature extraction, the curve fluctuates more noticeably when iteration counts range from 8 to 11, also achieving the best model effect at iteration count 7, with performance declining significantly after reaching 11 iterations. Additionally, the figure shows that model performance in both non-robust feature extraction processes follows a similar trend with step size changes, first increasing then decreasing. The optimal step size for structural non-robust feature extraction is 5e-5, while the optimal step size for attribute non-robust feature extraction is 5e-4.

---

4 Conclusion

This paper proposes an ensemble adversarial training strategy for graph convolutional neural networks based on non-robust features. By extracting non-robust features from embedding vectors after graph convolutional layers for adversarial training, this strategy bypasses issues such as data discreteness faced when directly constructing adversarial samples. To fully utilize graph data information, the proposed strategy extracts non-robust features from both topology structure and node attribute perspectives using random graphs, performs adversarial training on two sub-models with these non-robust features, and finally integrates the embedding vectors from both sub-models to obtain node prediction classifications.

Experiments on citation networks demonstrate that on the original Cora, Citeseer, and PubMed datasets, the proposed strategy improves accuracy by 0.84%, 1.25%, and 0.32% respectively compared to current best models. On the Cora dataset, when facing adversarial attacks with 20% perturbation rate, it improves accuracy by 6.91% compared to existing best models. These results fully prove that the proposed strategy can improve model performance on node classification tasks for both clean and attacked graphs.

Comparison between the ensemble model and single feature learning models shows that the strategy integrating both structural topology and node attributes achieves better results than models trained on only one aspect, both on original datasets and under attack scenarios.

Future work will focus on improving non-robust feature extraction effectiveness for datasets containing many isolated nodes, investigating the sensitivity and learning performance of other graph neural network model structures to non-robust features, and more deeply exploring the relationship between non-robust features in graph data and adversarial attacks.

References

[1] Zügner D, Akbarnejad A, Günnemann S. Adversarial attacks on neural networks for graph data [C]// Proc of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2018: 2847-2856.

[2] Dai Hanjun, Li Hui, Tian Tian, et al. Adversarial attack on graph structured data [C]// Proc of the 35th International Conference on Machine Learning. [S. l.]: PMLR Press, 2018: 1115-1124.

[3] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples [C]// Proc of the 6th ICLR, 2015.

[4] Pang Tianyu, Xu Kun, Du Chao, et al. Improving adversarial robustness via promoting ensemble diversity [C]// Proc of the 36th International Conference on Machine Learning. [S. l.]: PMLR Press, 2019: 4970-4979.

[5] Kariyappa S, Qureshi M K. Improving adversarial robustness of ensembles with diversity training [EB/OL]. (2019-01-28) [2022-01-04]. https://arxiv.org/pdf/1901.09981.

[6] Wu Xuguang, Wu Huijun, Zhou Xu, et al. CoG: a two-view co-training framework for defending adversarial attacks on graph [EB/OL]. (2021-9-12) [2022-01-04]. https://arxiv.org/pdf/2109.05558.

[7] Sun Lichao, Dou Yingtong, Yang C, et al. Adversarial attack and defense on graph data: A survey [EB/OL]. (2020-07-12) [2022-01-04]. https://arxiv.org/pdf/1812.10528.

[8] Dai Quanyu, Shen Xiao, Zhang Liang, et al. Adversarial training methods for network embedding [C]// Proc of WWW Conference. New York: ACM Press, 2019: 329-339.

[9] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations [C]// Proc of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2014: 701-710.

[10] Feng Fuli, He Xiangnan, Tang Jie, et al. Graph adversarial training: Dynamically regularizing based on graph structure [J]. IEEE Trans on Knowledge and Data Engineering, 2019, 33(6): 2493-2504.

[11] Wang Xiaoyun, Liu Xuanqing, Hsieh C J. GraphDefense: Towards robust graph convolutional networks [EB/OL]. (2019-11-11) [2022-01-04]. https://arxiv.org/pdf/1911.04429.

[12] Ilyas A, Santurkar S, Tsipras D, et al. Adversarial examples are not bugs, they are features [J]. Advances in Neural Information Processing Systems, 2019, 32: 125-136.

[13] Tramèr F, Papernot N, Goodfellow I, et al. The space of transferable adversarial examples [EB/OL]. (2017-05-23) [2022-01-04]. https://arxiv.org/pdf/1704.03453.

[14] Zhang Jiajie, Guo Yi, Wang Jiahui, et al. Ensemble learning framework for graph neural network with feature and structure enhancement [J/OL]. Application Research of Computers, 2021, 39(3). (2021-12-07) [2022-01-04]. https://www.arocmag.com/article/02-2022-03-033.html.

[15] Yang Huanrui, Zhang Jingyang, Dong Hongliang, et al. DVERGE: diversifying vulnerabilities for enhanced robust generation of ensembles [J]. Advances in Neural Information Processing Systems, 2020, 33: 4970-4979.

[16] Garg S, Sharan V, Zhang B H, et al. A spectral view of adversarially robust features [J]. Advances in Neural Information Processing Systems, 2018, 31: 10159–10169.

[17] Jin Wei, Ma Yao, Liu Xiaorui, et al. Graph structure learning for robust graph neural networks [C]// Proc of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2020: 66-74.

[18] Zügner D, Günnemann S. Adversarial attacks on graph neural networks via meta learning [EB/OL]. (2019-02-22) [2022-01-04]. https://arxiv.org/pdf/1902.08412.

[19] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [EB/OL]. (2017-02-22) [2022-01-04]. https://arxiv.org/pdf/1609.02907.

[20] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks [C]// Proc of the 6th ICLR, 2018.

[21] Jin Wei, Derr T, Wang Yiqi, et al. Node similarity preserving graph convolutional networks [C]// Proc of the 14th ACM International Conference on

Web Search and Data Mining. New York: ACM Press, 2021: 148-156.