Provably Robust Node Classification via Low-Pass Message Passing

Yiwei Wang*, Shenghua Liu†, Minji Yoon‡, Hemank Lamba‡, Wei Wang*, Christos Faloutsos‡, Bryan Hooi*
*School of Computing, National University of Singapore, Singapore
Email: {y-wang,wangwei,bhooi}@comp.nus.edu.sg
†Institute of Computing Technology, Chinese Academy of Sciences, China
Email: liushenghua@ict.ac.cn
‡Carnegie Mellon University, United States
Email: {minjiy,hlamba,christos}@cs.cmu.edu

***Abstract*—Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance on node classification. However, recent works have shown that GCNs are vulnerable to adversarial attacks, such as additions or deletions of adversarially-chosen edges in the graph, in order to mislead the node classification algorithms. How can we design robust GCNs that are resistant to such adversarial attacks? More challengingly, how can we do this in a way that is provably robust? We propose a robust node classification approach based on a low-pass 'message passing' mechanism, that (a) reduces the effectiveness of adversarial attacks in experiments, and (b) provides theoretical guarantees against adversarial attacks. Our approach can be embedded into the existing GCN architectures to enhance their robustness. Empirical results show that our loss-pass method effectively improves the performance of multiple GCNs under miscellaneous perturbations and helps them to achieve superior performance on various graphs.**

***Keywords*-Graph Convolutional Networks, Node Classification, Robustness**

I. INTRODUCTION

Node classification is a fundamental task on graph data, which is to classify the nodes in an (attributed) graph [1]: for example, predicting protein types in a protein interaction graph [2]. In recent years, graph neural network methods such as Graph Convolutional Networks (GCNs) have achieved state-of-the-art performance for node classification [3]–[5], leading to tremendous interest.

Despite their effectiveness for node classification, GCNs have been found to be vulnerable to adversarial attacks [6]. This is a significant drawback in terms of their reliability in many real-world settings, particularly in risk-sensitive scenarios such as security, healthcare, and finance [7]. It is thus essential to design GCNs that are robust against adversarial attacks.

GCNs apply a 'message passing' mechanism to make predictions, whereby they aggregate semantic representations of each node and its neighbors at each layer. On a clean graph structure, this message passing process tends to produce the similar predictions to the connected nodes [8]. However, when adversarial edges are present, it incurs over-whelming erroneous information and misleads the predictions heavily.

In this paper, we propose a **low-pass 'message passing'** mechanism that provides higher robustness for GCNs without the loss of effectiveness. The mechanism weakens 'message passing' between semantically dissimilar nodes. We design this mechanism based on the observation that node pairs which are very semantically different are the most susceptible to being attacked by the adversary, as perturbations across such pairs have a stronger effect on the prediction results. Therefore, by weakening the 'message passing' between dissimilar nodes, we inhibit the prediction deviations induced by adversarial attacks. We weaken the 'message passing' strength of an adversarial edge to the extent that is positively related to the distance between the semantic representations of the nodes on its ends, because the more different the nodes linked by an adversarial edge are, the more deviations on the predictions are induced by the edge (see Eq. (2)).

Recent research in adversarial machine learning has seen a rapid back-and-forth between adversarial attacks and defenses, in which several advanced defense approaches were broken by newly proposed attack methods [9]. Such 'arms races' leave machine learning systems vulnerable to attack. To circumvent this problem, our solution is to provide *theoretical guarantees* for the robust node classification beyond the intuitive justification, whereby the model output does not excessively change when faced with adversarial perturbations. In particular, we focus on defending against structural perturbations, i.e., adding/deleting edges, which fit well for the characteristics of graph data.

To provide the provable robustness guarantees, a challenge is that the complex non-linear activation functions in GCNs make the relations between inputs and outputs hard to analyze. The irregular and noisy graph structure aggravates this challenge. Another challenge lies in the 'message passing' of GCNs, with which the predictions of widespread nodes would be affected by even a small number of adversarially added/deleted edges. Therefore, we have to consider widespread nodes and edges for analyzing the prediction of a node. To address these challenges, we

Shenghua Liu is also with CAS Key Laboratory of Network Data Science & Technology, CAS, and University of Chinese Academy of Sciences, Beijing 100049, China.

first derive the deviation bound for a single layer. This enables us to consider only the effects within the first-hop neighborhood, where we mainly use Lipschitz conditions [10] and the Triangle Inequality [11] for analysis. Next, we extend the bound to the case of multiple stacked layers, and finally obtain the error bounds for the model predictions in Theorem 2 with our low-pass 'message passing' mechanism.

Our low-passing 'message passing' is a general approach that can be incorporated into the existing GCN architectures, e.g., JKNet [12], IncepGCN [13], ResGCN [14], to enhance their robustness. We conduct extensive experiments over the public graph datasets: Cora, Citeseer, Pubmed [1], Coauthor-CS and Coauthor-Physics [15] to evaluate the efficacy of our proposed method on the task of node classification. The empirical results show that, under various adversarial attacks, e.g., RL-S2V, [16], NETTACK [6], etc., our low-pass approach effectively improves the performances of multiple GCNs and enable them to consistently outperform benchmark methods in accuracy by 5%-20%. Furthermore, we observe that the derived guarantee on error bounds is reliable even when most nodes in the graph are attacked.

Our contributions are as follows:

- **Low-Pass Message Passing**: We propose a novel approach for making GCNs robust against adversarial edge additions and deletions, by weakening 'message passing' between the semantically dissimilar nodes.
- **Provable Robustness**: In Theorem 2, we provide formal robustness guarantees against graph structural attacks with our low-pass 'message passing' mechanism.
- **Effectiveness**: Our experimental results show that our approach outperforms benchmarks in accuracy by 5%-20% on graphs under adversarial attacks.

II. RELATED WORK

Graph neural networks have shown promising performance in graph learning [3], [17], [18], [19], [20]. More recently, many works have shown that GCNs are susceptible to adversarial evasion and poisoning attacks. Evasion attacks assume that during the testing phase, the adversary is allowed to add edges to each node. Poisoning attacks, in contrast, are done at training time [21], i.e., the attacks exist in both training and testing stages. [22] provides an early survey of this area. [23] gives a meta-learning approach to learn poisoning attacks on node classification models.

However, works on the defensive side, i.e., graph neural network approaches that are robust against adversaries, are sparser. [24] proposes an adversarial training approach for graph neural networks. However, this work focuses on a particular type of attack (gradient-based attacks) and does not provide theoretical results. In [21], the authors propose a robust GCN model, which tries to defend against poisoning attacks by using Gaussian distributions as the hidden representations to replace plain vectors. However, they provide no robustness guarantees, and our low pass 'message passing', which follows the plain vectors of standard GCNs, is more succinct and leads to better generalization and interpretability. [25] improves the robustness by removing existing edges whose connected nodes have low feature similarities.

Our proposed method differs from existing methods in the sense that: it presents a new 'message passing' module that can be embedded into diverse GCN models for higher robustness instead of a brand new GCN architecture; we provide the provable bound on the prediction deviations with attacked graph structures.

III. PROBLEM DEFINITION

In this section, we introduce the essential preliminaries and notations of the graph data, GCN models, as well as the corresponding attacks and our target.

A. Preliminaries

We define a graph as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertex set consisting of N vertices, and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the set of edges. Let $\mathbf{A}$ be the $N \times N$ adjacency matrix of the given graph, $A_{ij} = 1$ indicates that edge e_{ij} exists between node i and node j, and $A_{ij} = 0$, otherwise.

Let $\mathcal{N}_i$ be the neighborhood of node i (including itself) and $|\mathcal{N}_i|$ counts the cardinality of $\mathcal{N}_i$. Given a vector $\mathbf{v}$, we use $\|\mathbf{v}\|$ to denote its ℓ_1 norm; while given a matrix $\mathbf{X}$, we denote its ℓ_∞ norm as $\|\mathbf{X}\|$.

B. Graph Convolutional Networks

We introduce the well-established multi-layer GCN model in [3] for node classification, which first adds self-loops to the adjacency matrix in $\tilde{\mathbf{A}} = \mathbf{A}+\mathbf{I}$, then defines the $(l+1)$-th layer by:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \tag{1}$$

Here $\mathbf{W}^{(l)}$ is a trainable weight matrix for layer l, $\mathbf{D}$ is the diagonal matrix of degrees ($D_{ii} = \sum_j \tilde{A}_{ij} = d_i$), and σ is the non-linear activation function ReLU [26]. The first layer is defined using node features $\mathbf{X}$ as input, i.e., $\mathbf{H}^{(0)} = \mathbf{X}$ (one row per node). In the output layer, the hidden representation $\mathbf{H}^{(L)}$ is passed through a Softmax function to obtain predicted categories for each node: $\mathbf{P} = softmax(\mathbf{H}^{(L)})$. Note that GCNs in practice are typically limited to 2 or 3 layers for node classification [3]. Note that $\mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}$ is the aggregation matrix for governing the 'message passing' across GCN layers.

C. Adversarial Attack and Our Target

Although the GCN model defined in Eq. (1) is effective for node classification, existing works [23] have confirmed that adversarial attacks can heavily degrade its performance.

The adversary adds or removes any edges for the targeted graph nodes in order to affect their classification output as much as possible, subject to a set of **budget** constraints. Let $\tilde{\mathcal{N}}_i$ be the perturbed neighborhood of node i. The budget

of the adversary is measured in terms of the fraction of each node's edges that the adversary can modify. Let $\mathcal{M}_i = (\mathcal{N}_i \setminus \check{\mathcal{N}}_i) \cup (\check{\mathcal{N}}_i \setminus \mathcal{N}_i)$ represent the set of neighbors added or deleted by the adversary to node i. The budget constraint is $|\mathcal{M}_i|/d_i < \Delta$ for all i, where Δ is the given budget limit.

Our target is to propose a robust node classification method that effectively defends against graph structural attacks. Additionally, we focus on obtaining an upper bound for the deviations of the log probability $|\log \check{P}_{ic} - \log P_{ic}|$ of *any* node i and *any* class c under *any* perturbation. This bound limits the damage that future intelligent attacks may cause over GCNs with our low-pass 'message passing'. Consequently, we resolve the issue that attacks proposed in the future can easily break currently 'robust' models.

IV. LOW-PASS MESSAGE PASSING

In this section, we introduce our low-pass 'message passing' mechanism. We first present our low-pass aggregation matrix for controlling the message passing across GCN layers and explain why GCNs are robust with it. Then, we prove theoretical guarantees for the robustness. Finally, we present the complexity analysis.

A. Low-Pass Aggregation Matrix

Recall that in Eq. (1), $\mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}$ acts as the aggregation matrix governing 'message passing'. Denote $\mathbf{h}_i^{(l)}$ as the l-th layer embedding of node i, i.e., the i-th row of $\mathbf{H}^{(l)}$. In a GCN layer, the aggregated representation of node i can be expressed as $\mathbf{h}_i^{aggre} = \sum_{j\in\mathcal{N}_i} \frac{1}{\sqrt{d_i d_j}}\mathbf{h}_j$ by symmetric aggregation as introduced in Eq. (1) or $\mathbf{h}_i^{aggre} = \sum_{j\in\mathcal{N}_i} \frac{1}{d_j}\mathbf{h}_j$ by row normalization. In practice, these two expressions give similar performance [27]. With row normalization, we have:

$$\mathbf{h}_i^{aggre} = \sum_{j\in\mathcal{N}_i} \frac{1}{d_i}\mathbf{h}_j = \mathbf{h}_i + \frac{1}{d_i}\sum_{j\in\mathcal{N}_i\setminus\{i\}} (\mathbf{h}_j - \mathbf{h}_i) \quad (2)$$

For $\mathbf{h}_i^{aggre}$, $\mathbf{h}_i$ is independent from the graph structure. Neighbor j influences $\mathbf{h}_i^{aggre}$ by $\mathbf{h}_j - \mathbf{h}_i$. The 'message passing' strength from different neighbors (edge weight) are equal. This mechanism leaves large space for the adversarial attacks, since through the fake edge e_{ij}, for example, the perturbations $\mathbf{h}_j - \mathbf{h}_i$, which is potential to be excessively large, can be propagated to node i ($\mathbf{h}_i^{aggre}$). For the robustness of GCNs, we aim to limit the influence that a node can have on another. Denote $R > 0$ as the threshold for controlling our low-pass 'message passing'. We define the weight of edge e_{ij} at layer l as:

$$\beta_{ij}^{(l)} = \frac{R}{\max(R, \|\mathbf{h}_i^{(l)} - \mathbf{h}_j^{(l)}\|)} \quad (3)$$

Note that this assigns a weight of 1 if $\mathbf{h}_i^{(l)}$ and $\mathbf{h}_j^{(l)}$ are less than R apart, while gradually reducing the weight as the distance between them exceeds R. Then, our edge weights $\beta_{ij}^{(l)}$ acts as a low-pass filter to prevent any node from being excessively affected by any of its neighbors, and vice versa. In addition, since $\mathbf{h}_j^{(l)} - \mathbf{h}_i^{(l)}$ infers how different nodes i and j are semantically, $\beta_{ij}^{(l)}$ weakens the strength of the edges connecting significantly distinct nodes, which are likely to be adversarial. An example of the effects of our $\beta_{ij}^{(l)}$ is shown in Fig. 1. Intuitively, utilizing $\beta_{ij}^{(l)}$ inhibits the influence of adversarial edges.

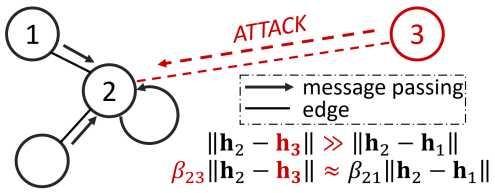


Figure 1: **Our low-pass edge weights inhibit the effects from adversarial edges.** Edge e_{23} is injected for attacking node 2. With the original message passing (see. Eq. (2)), the feature $\mathbf{h}_3$ affects $\mathbf{h}_2^{aggre}$ heavily by $\mathbf{h}_3 - \mathbf{h}_2$. But our low-pass edge weight β_{23} inhibits this effect.

Based on the above analysis, we define our low-pass aggregation matrix $\mathbf{B}^{(l)}$ on layer l to replace $\mathbf{D}^{-1/2}\tilde{\mathbf{A}}\mathbf{D}^{-1/2}$:

$$B_{ij}^{(l)} = \begin{cases} \beta_{ij}^{(l)}/d_i & \text{if } j \in \mathcal{N}_i \setminus \{i\} \\ 1 - \sum_{j\in\mathcal{N}_i\setminus\{i\}} \beta_{ij}^{(l)}/d_i & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

For each edge e_{ij}, we set $B_{ij}^{(l)} = \beta_{ij}^{(l)}/d_i$ on the l-th layer. This definition could result in the entries of row i summing to less than 1. Hence, we add the necessary amount to the diagonal entry $B_{ii}^{(l)}$ to make the row sum to 1: i.e., $B_{ii}^{(l)} = 1 - \sum_{j\in\mathcal{N}_i\setminus\{i\}} \beta_{ij}^{(l)}/d_i$. We add more strengths for the self-loops because the features belonging to a node itself is more essential than those from other nodes.

A unified view of GCN with our low-pass 'message passing' is shown on the left side of Figure 2. The right side of Figure 2 shows the final-layer representations given by the original GCN, compared with our low-pass approach. After injecting the adversarial edge e_{12}, we observe a large deviation in node 1's embedding. However, our low-pass 'message passing' restrain the deviation greatly, which implies better robustness.

Note that our low-pass 'message passing' mechanism is to use the aggregation matrix $\mathbf{B}$ defined in Eq. (4) to conduct the message passing in GCN operations. Hence, it is applicable to the general existing GCN architectures, e.g., JKNet [12], IncepGCN [13], ResGCN [14]. These advanced GCN variants are shown to be more effective than the vanilla GCN to some extent, and our approach can additionally enhance their robustness under perturbations.

B. Provable Robustness

Our theoretical results show for any perturbation perturbations under a given budget, our approach provides a limit

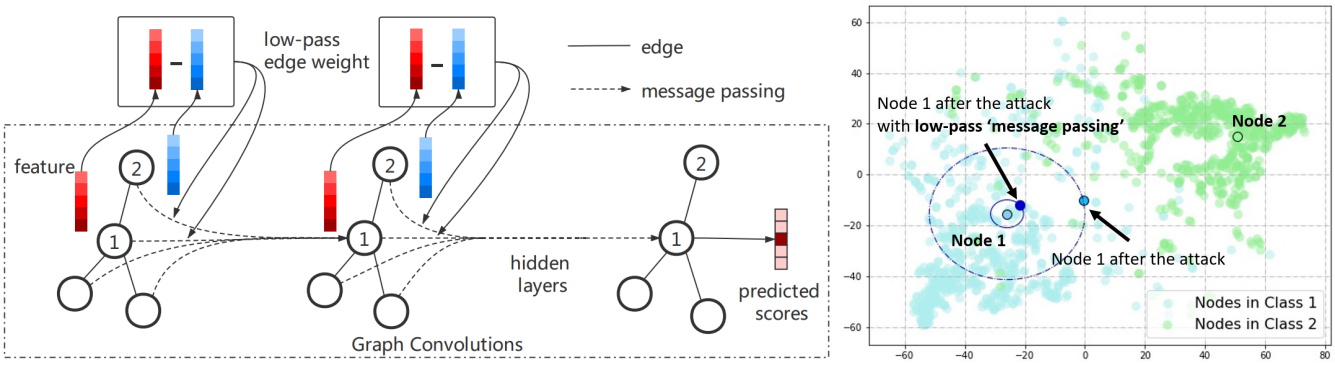


Figure 2: (*left*) **GCN with our low-pass 'message passing'.** The low-pass edge weight between nodes 1 and 2 is computed based on their representations. (*right*) The final-layer hidden representations of the nodes retrieved by GCN belonging to two classes in Citeseer [1] (visualized via t-SNE [28]). After the adversarial injection of edge e_{12}, severe deviations happen on node 1. But with our low-pass 'message passing', its deviation is inhibited.

on how much the deviations are produced on predictions. This is possible due to our edge weighting function $\beta_{ij}^{(l)}$, limits the effects of each edge. We first show a lemma to account for the effect of ReLU , i.e., $\sigma(\cdot)$:

Lemma 1. *Given any matrices* $\mathbf{X}, \mathbf{Y}$*, we have*

$$\|\mathbf{X}-\mathbf{Y}\| \geq \|\sigma(\mathbf{X})-\sigma(\mathbf{Y})\| \tag{5}$$

Proof: Recall that a function f is Lipschitz with constant C if $\frac{|f(x)-f(y)|}{|x-y|} \leq C$ for all x and y. Note that $f(x) := |x|$ is Lipschitz with constant 1 by triangle inequality, so that $\sigma(x) = (x+|x|)/2$ is Lipschitz with constant 1 as well. Applying this elementwise, $\mathbf{X}-\mathbf{Y}$ has elementwise larger absolute values than $\sigma(\mathbf{X})-\sigma(\mathbf{Y})$. Thus, by definition of the norm we have $\|\mathbf{X}-\mathbf{Y}\| \geq \|\sigma(\mathbf{X})-\sigma(\mathbf{Y})\|$. ■

Next, we will bound the deviations on the $(l+1)$-th layer ($\|\mathbf{H}^{(l+1)} - \check{\mathbf{H}}^{(l+1)}\|$) as a function of the deviations of the l-th layer ($\|\mathbf{H}^{(l)} - \check{\mathbf{H}}^{(l)}\|$). We then use it to derive a bound for $\|\mathbf{H}^{(l)} - \check{\mathbf{H}}^{(l)}\|, \forall l \geq 0$. Recalling that the set of the adversarial edges is $\mathcal{M}_i = (\mathcal{N}_i \setminus \check{\mathcal{N}}_i) \cup (\check{\mathcal{N}}_i \setminus \mathcal{N}_i)$, we have:

Lemma 2. *Given the budget of the graph structural attacks as* $\mathcal{M}_i/d_i \leq \Delta < 1, \forall i$*, the deviation of* $\mathbf{H}^{(l+1)}$ *is bounded by:*

$$\|\mathbf{H}^{(l+1)} - \check{\mathbf{H}}^{(l+1)}\| \leq \|\mathbf{W}^{(l)}\|\left(\frac{2R\Delta}{1+\Delta} + 4\|\mathbf{H}^{(l)} - \check{\mathbf{H}}^{(l)}\|\right)$$

Proof:

$$\begin{aligned}
&\|\mathbf{H}^{(l+1)} - \check{\mathbf{H}}^{(l+1)}\| \\
=&\|\sigma(\mathbf{B}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) - \sigma(\check{\mathbf{B}}^{(l)}\check{\mathbf{H}}^{(l)}\mathbf{W}^{(l)})\| \\
\leq&\|\mathbf{B}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)} - \check{\mathbf{B}}^{(l)}\check{\mathbf{H}}^{(l)}\mathbf{W}^{(l)}\| \quad \text{by Lemma 1} \\
\leq&\|\mathbf{B}^{(l)}\mathbf{H}^{(l)} - \check{\mathbf{B}}^{(l)}\check{\mathbf{H}}^{(l)}\| \cdot \|\mathbf{W}^{(l)}\|.
\end{aligned}$$

The norm of row i of the matrix $\mathbf{B}^{(l)}\mathbf{H}^{(l)} - \check{\mathbf{B}}^{(l)}\check{\mathbf{H}}^{(l)}$ is:

$$\|\sum_{j\in\mathcal{N}_i}\frac{\beta_{ij}^{(l)}}{d_i}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}) - \sum_{j\in\check{\mathcal{N}}_i}\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|$$

$$\leq\|\sum_{j\in\mathcal{N}_i}\frac{\beta_{ij}^{(l)}}{d_i}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}) - \sum_{j\in\check{\mathcal{N}}_i}\frac{\check{\beta}_{ij}^{(l)}}{d_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\| + \tag{6}$$

$$\|\sum_{j\in\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}}{d_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}) - \sum_{j\in\check{\mathcal{N}}_i}\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\| \tag{7}$$

We first consider the edge addition case. Denote a_i as the number of edges added to node i. The term in Eq. (7) is bounded as follows:

$$\begin{aligned}
&\left\|\sum_{j\in\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}}{d_i}\left(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\right) - \sum_{j\in\check{\mathcal{N}}_i}\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}\left(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\right)\right\| \\
=&\|\sum_{j\in\check{\mathcal{N}}_i\setminus\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}) - \sum_{j\in\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}a_i}{d_i\cdot\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\| \\
\leq&\sum_{j\in\check{\mathcal{N}}_i\setminus\mathcal{N}_i}\|\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\| + \\
&\sum_{j\in\mathcal{N}_i}\|\frac{\check{\beta}_{ij}^{(l)}a_i}{d_i\cdot\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\| \\
\leq&\sum_{j\in\check{\mathcal{N}}_i\setminus\mathcal{N}_i}\frac{R}{\check{d}_i} + \sum_{j\in\mathcal{N}_i}\frac{Ra_i}{d_i\cdot\check{d}_i} \\
=&\frac{2R\cdot a_i}{\check{d}_i} \\
=&2R\frac{\Delta}{1+\Delta}
\end{aligned}$$

For the term in Eq. (6), we have:

$$\begin{aligned}&\|\sum_{j\in\mathcal{N}_i}\frac{\beta_{ij}^{(l)}}{d_i}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\sum_{j\in\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}}{d_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\=&\sum_{j\in\mathcal{N}_i}\frac{1}{d_i}\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\end{aligned}$$

Let $D=2\|\mathbf{H}^{(l)}-\check{\mathbf{H}}^{(l)}\|$. Then $D\geq\|(\mathbf{h}_i^{(l)}-\mathbf{h}_j^{(l)})-(\check{\mathbf{h}}_i^{(l)}-\check{\mathbf{h}}_j^{(l)})\|$ for any j. Denote $m=\max\left(\|\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}\|,R\right),\check{m}=\max\left(\|\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\|,R\right)$. For any index j, we have two cases:

case 1: $\|\mathbf{h}_i^{(l)}-\mathbf{h}_j^{(l)}\|\geq\|\check{\mathbf{h}}_i^{(l)}-\check{\mathbf{h}}_j^{(l)}\|$. Then:

$$\begin{aligned}&\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\beta_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|+\\&\|\beta_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&\beta_{ij}^{(l)}\cdot D+\|\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\||\beta_{ij}^{(l)}-\check{\beta}_{ij}^{(l)}|\\\leq&D+\|\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\|(\frac{R}{\check{m}}-\frac{R}{\check{m}+D})\\=&D+\|\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)}\|\frac{RD/\check{m}}{\check{m}+D}\\\leq&2D\end{aligned}$$

case 2: $\|\mathbf{h}_i^{(l)}-\mathbf{h}_j^{(l)}\|<\|\check{\mathbf{h}}_i^{(l)}-\check{\mathbf{h}}_j^{(l)}\|$. Then:

$$\begin{aligned}&\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})\|+\\&\|\check{\beta}_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&\|\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}\||\beta_{ij}^{(l)}-\check{\beta}_{ij}^{(l)}|+\check{\beta}_{ij}^{(l)}\cdot D\\\leq&\|\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}\|(\frac{R}{m}-\frac{R}{m+D})+D\\=&\|\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)}\|\frac{RD/m}{m+D}+D\\\leq&2D\end{aligned}$$

We conclude that Eq. (6) is bounded by

$$\begin{aligned}&\|\sum_{j\in\mathcal{N}_i}\frac{\beta_{ij}^{(l)}}{d_i}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\sum_{j\in\mathcal{N}_i}\frac{\check{\beta}_{ij}^{(l)}}{d_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&\sum_{j\in\mathcal{N}_i}\frac{1}{d_i}\|\beta_{ij}^{(l)}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\check{\beta}_{ij}^{(l)}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&2D\end{aligned}$$

Finally, combining the bounds on Eq. (6) and (7) gives:

$$\begin{aligned}&\|\sum_{j\in\mathcal{N}_i}\frac{\beta_{ij}^{(l)}}{d_i}(\mathbf{h}_j^{(l)}-\mathbf{h}_i^{(l)})-\sum_{j\in\check{\mathcal{N}}_i}\frac{\check{\beta}_{ij}^{(l)}}{\check{d}_i}(\check{\mathbf{h}}_j^{(l)}-\check{\mathbf{h}}_i^{(l)})\|\\\leq&2D+2R\frac{\Delta}{1+\Delta}\\=&4\|\mathbf{H}^{(l)}-\check{\mathbf{H}}^{(l)}\|+2R\frac{\Delta}{1+\Delta}.\end{aligned}$$

In the case of both edge deletions and additions, we have a similar bound as

$$4\|\mathbf{H}^{(l)}-\check{\mathbf{H}}^{(l)}\|+2R\frac{\Delta}{1+\Delta},$$

of which the derivation is similar to the edge addition case. Overall, we have the bound in (2). ■

Finally, this allows us to bound the adversarial perturbation on the final layer:

Theorem 1. *Given the attack budget: $\mathcal{M}_i/d_i\leq\Delta<1$, we have:*

$$\|\mathbf{H}^{(L)}-\check{\mathbf{H}}^{(L)}\|\leq 2R\frac{\Delta}{1+\Delta}(\sum_{s=0}^{L-1}\|\mathbf{W}^{(s)}\|\prod_{l=s+1}^{L-1}4\|\mathbf{W}^{(l)}\|).$$

Proof: This follows from repeatedly applying Lemma 2 to each successive layer of the network. ■

Using this result, we show that the predicted probability for each node can only change by a small factor. Let C be the number of classes, and P_{ic} be the predicted probability of node i belonging to class c. We bound the prediction deviations as:

Theorem 2. *The adversarial perturbation in the log-probability for node i belonging to any class c is bounded by:*

$$\begin{aligned}&|\log\check{P}_{ic}-\log P_{ic}|\\\leq&\|\mathbf{H}^{(L)}-\check{\mathbf{H}}^{(L)}\|\\\leq&2R\frac{\Delta}{1+\Delta}(\sum_{s=0}^{L-1}\|\mathbf{W}^{(s)}\|\prod_{l=s+1}^{L-1}4\|\mathbf{W}^{(l)}\|)\end{aligned}\tag{8}$$

Proof: We denote $c_{max}=\arg\max_{c'}\{H_{ic'}^{(L)}-\check{H}_{ic'}^{(L)}\}$. By the definition of the softmax function, we have

$$\begin{aligned}\check{P}_{ic}/P_{ic}&=e^{\check{H}_{ic}^{(L)}-H_{ic}^{(L)}}\frac{\sum_{c'=1}^{C}e^{H_{ic'}^{(L)}}}{\sum_{c'=1}^{C}e^{\check{H}_{ic'}^{(L)}}}\\&\leq e^{\check{H}_{ic}^{(L)}-H_{ic}^{(L)}}\frac{e^{H_{ic_{max}}^{(L)}}}{e^{\check{H}_{ic_{max}}^{(L)}}}\\&=e^{\left(\check{H}_{ic}^{(L)}-H_{ic}^{(L)}\right)+\left(H_{ic_{max}}^{(L)}-\check{H}_{ic_{max}}^{(L)}\right)}\\&\leq e^{\|\mathbf{H}^{(L)}-\check{\mathbf{H}}^{(L)}\|}\end{aligned}\tag{9}$$

If $c=c_{max}$, the step (9) holds with $\check{H}_{ic}^{(L)}-H_{ic}^{(L)}+H_{ic_{max}}^{(L)}-\check{H}_{ic_{max}}^{(L)}=0$. Otherwise, it holds because

Table I: Statistics of the utilized datasets

Dataset	# Nodes	# Edges	# Classes	# Features
Cora	2,708	5,429	7	1,433
Citeseer	3,327	4,732	6	3,703
Pubmed	19,717	44,338	3	500
Co-CS	18,333	81,894	67	6,805
Co-Phy	34,493	247,962	5	8,415

$\left|\check{H}_{ic}^{(L)} - H_{ic}^{(L)}\right| + \left|H_{ic_{max}}^{(L)} - \check{H}_{ic_{max}}^{(L)}\right| \leq \|\mathbf{H}^{(L)} - \check{\mathbf{H}}^{(L)}\|$.This derivation can be extended to the lower bound trivially: $\check{P}_{ic}/P_{ic} \geq e^{-\|\mathbf{H}^{(L)} - \check{\mathbf{H}}^{(L)}\|}$. Thus, we have the upper bound for the log-probability deviations as $\|\mathbf{H}^{(L)} - \check{\mathbf{H}}^{(L)}\|$. Then, Eq. (8) follows with Theorem 1. ■

When $\Delta \to 0$ or $R \to 0$, the error bounds presented in both Theorem 1 and 2 approach 0 as well, which match the intuitions that fewer perturbations induce less deviation, and smaller R prevents more attacks. The bound grows with $\|\mathbf{W}^{(l)}\|$, which which agrees with our prior knowledge that perturbations are amplified by matrix multiplication with the GCN parameter matrices. Note that for the task of node classification, the layer number of GCNs, i.e., L, is limited to 2 or 3 [3]. As a result, the value in the brackets in Eq. (8) is small. For example, when $L = 2$ and $\|\mathbf{W}^{(l)}\| = 1$ holds, Eq. (8) is equal to:

$$2R\frac{\Delta}{1+\Delta}(4+1) = 10R\frac{\Delta}{1+\Delta} \leq 10R\Delta \qquad (10)$$

Thus, in practice, we have the bound obtained from Eq. (8) as $O(R\Delta)$, which is linear in both the hyper-parameter R and the attack budget Δ.

C. Complexity Analysis

Compared with the original 'message passing', our low-pass approach needs to additionally compute the weights for each edge, as shown in Eq. (3), leading to the time complexity of $O(|\mathcal{E}|\sum_{i=1}^{L} c_i)$, where c_i is the dimension of layer i. This is smaller than the time complexity of GCN [3]. Thus, our low-pass 'message passing' does not change the time complexity of GCN models.

V. EXPERIMENTS

In this section, we present the empirical results of our proposed approach to the task of node classification. We give the experimental settings first and then present and analyze the experimental results. Overall, we aim to answer the following questions:
(1) **Adversarial Robustness:** Does our method outperform baselines in robustness against adversarial attacks?
(2) **Theoretical Guarantees:** Are our theoretical guarantees reliable under perturbations?
(3) **Parameter Sensitivity:** Is our approach sensitive to hyper-parameters?

A. Experimental Settings

Our baselines are as follows:

- **GCN** [3] is one of the most commonly used approaches for node classification.
- **JKNet** [12] add skip connections between GCN hidden layers to the top layer to address the over-smoothing problem [32].
- **IncepGCN** [13] uses the structure of inception networks [33] to organize GCN layers.
- **GAT** [29] introduces the self-attention mechanism to enhance the learning capacity.
- **RGCN** [21] uses Gaussian distributions in GCN layers to absorb adversarial effects.
- **PreProcess** [25] improves the robustness by removing existing edges whose connected nodes have low feature similarities.
- **PA-GNN** [30] uses meta-optimization to transfer the alleviation ability from clean graphs to the targeted poisoned graph.
- **Pro-GNN** [31] simultaneously learn the clean graph structure from perturbed graph and GNN parameters to defend against adversarial attacks

Note that RGCN, PreProcess, PA-GNN, and Pro-GNN are currently the state-of-the-art robust GCN methods. For the implementations of all the approaches, we follow the parameter settings suggested by the authors. For our low-pass 'message passing', we implement it within GCN, JKNet, and IncepGCN respectively, and set $R = 1$ by default, in which the hyper-parameters of GCN architectures are set as their authors suggest.

We focus on dealing with the attacks to graph structures, which are closely related to the graph data characteristics, as edge injections are often easy for an adversary to conduct, and of practical interest: e.g. fake reviews, fake links in social networks, etc [34]. We consider two types of standard attack settings: **evasion** (test-stage) and **poisoning** (train-stage) attacks. We use the following attack approaches:

- **Greedy** For each attacked node i with its predicted class y_i, we connect i with a specified number of nodes whose prediction scores for y_i are the lowest ones and who are not meanwhile neighbours of i.
- **RL-S2V** [16] is an effective adversarial attack method based on reinforcement learning, which is designed for evasion attacks.
- **NETTACK** [6] is an adversarial attack approach for GCNs, which focuses on solving a bi-level optimization problem. It is effective for poisoning attacks. We utilize the direct attack version of NETTACK, since it is more powerful than the indirect one.
- **Random** We remove/add edges between randomly selected node pairs.

Among these attack methods, the Greedy method is proposed by us for efficient poisoning and evasion attacks. For

Table II: Results of node classification with the random splits of 20 labeled examples per class, in terms of test accuracy (%). We report mean and standard derivations of 2000 trials on 20 splits with randomly selected targeted nodes.

Attack Method	Defense Method	Cora	Citeseer	Pubmed	Co-CS	Co-Phy
RL-S2V	GCN [3]	51.1±1.2	42.1±1.4	59.8±1.3	62.1±0.8	63.2±0.7
	JKNet [12]	52.7±1.4	44.1±1.1	61.1±0.7	61.7±1.2	62.8±1.5
	IncepGCN [13]	52.9±1.2	44.3±1.0	60.5±0.9	61.9±1.1	63.1±0.9
	GAT [29]	51.8±1.5	43.2±0.6	60.3±1.1	61.4±1.4	62.5±1.3
	RGCN [21]	58.6±1.3	52.8±1.4	66.5±1.6	68.3±1.9	69.5±2.1
	PreProcess [25]	61.3±2.3	57.2±1.2	69.8±1.4	72.9±1.5	73.1±0.9
	PA-GNN [30]	57.5±0.7	51.4±0.8	66.7±1.3	65.8±1.3	64.3±1.1
	Pro-GNN [31]	59.1±0.7	52.8±0.8	69.3±1.3	69.0±1.3	68.9±1.1
	Low-Pass GCN (Ours)	66.2±0.8	62.3±0.7	75.1±1.2	**77.1±0.7**	**79.2±0.9**
	Low-Pass JKNet (Ours)	66.7±0.7	62.5±0.9	**75.9±1.2**	76.9±0.6	79.0±1.2
	Low-Pass IncepGCN (Ours)	**66.9±0.7**	**62.8±0.9**	75.0±0.5	76.7±0.8	79.1±1.1
NETTACK	GCN [3]	49.2±1.4	44.8±1.2	60.7±1.1	61.1±1.1	61.4±0.7
	JKNet [12]	49.9±1.1	44.3±0.7	61.2±1.2	60.9±1.3	61.5±1.0
	IncepGCN [13]	49.8±1.1	45.2±1.3	60.9±0.7	61.0±0.8	61.1±1.1
	GAT [29]	49.3±1.3	44.2±1.1	60.2±0.5	60.3±1.7	61.2±1.1
	RGCN [21]	52.7±1.3	46.3±1.4	63.2±0.9	63.1±0.7	62.2±1.0
	PreProcess [25]	53.2±0.9	48.6±1.1	63.9±1.1	64.2±0.6	65.4±0.8
	PA-GNN [30]	57.9±1.1	52.1±1.3	68.4±1.4	68.4±1.1	67.6±1.5
	Pro-GNN [31]	61.2±1.4	56.2±0.8	71.9±1.2	73.7±0.9	72.1±1.1
	Low-Pass GCN (Ours)	64.3±0.8	60.2±1.1	74.3±0.9	**75.2±0.7**	**76.1±1.0**
	Low-Pass JKNet (Ours)	**64.8±0.6**	59.9±1.0	**74.5±0.8**	75.0±0.5	75.9±1.4
	Low-Pass IncepGCN (Ours)	64.5±0.5	**60.7±0.9**	74.2±1.3	75.0±0.8	75.4±1.3

the implementations of RL-S2V and NETTACK, we follow the default parameter settings in the authors' implementations.

We use standard benchmark datasets: Cora, Citeseer, Pubmed [1], Coauthor-CS (short as Co-CS) and Coauthor-Physics (short as Co-Phy) [15] for evaluation. The first three are citation networks, and the last two are co-author networks. Each of them contains an unweighted adjacency matrix and bag-of-words features. The statistics of these datasets are presented in Table I.

We conduct experiments on different datasets with random splits. We randomly select 20 nodes per class to form the training set, and the same number of samples for the validation. All the remaining ones are put into the testing set. We make 20 random splits for each dataset. And on each split, we conduct the experiments for a specific number of trials.

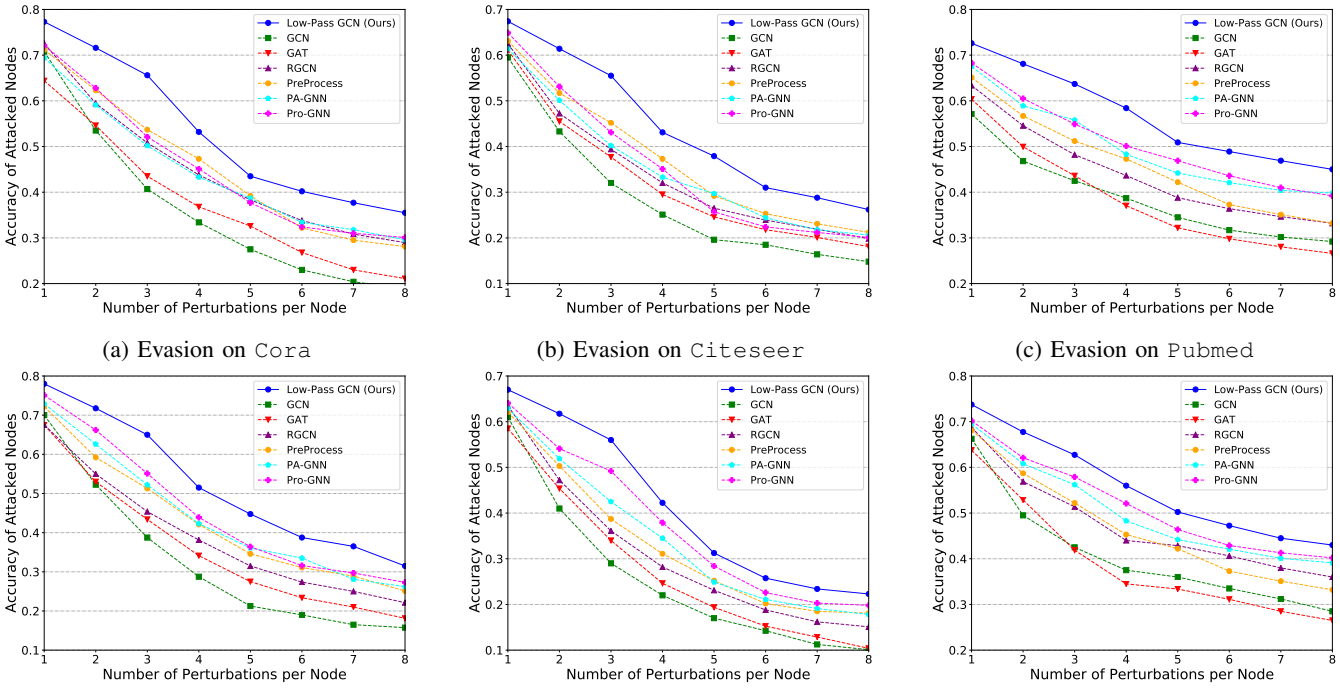
B. Experiments Against Adversarial Attacks

We consider both evasion and poisoning attacks in the experiments. In the former setting, the trainable weights in the model are kept unchanged after attacks, while in the latter one, the model is retrained after attacks. Under both attacking scenarios, we conduct the experiments for 100 trials with 50 nodes in the test set selected randomly as the targeted nodes that are to be attacked. This setting can help to observe the global performance of the defense models on different nodes.

We present the performance of different defense approaches given the attacking methods, RL-S2V (for evasion) and NETTACK (for poisoning), in Table II, where we perturb two edges per node on average. Our low-pass 'message passing' significantly improves the robustness of various GCN models and helps them to outperform the benchmark approaches on different datasets by 5%-20%. GAT is an attention GCN model designed as a strong alternative for GCN for node classification. But on the attacked graph structures, it performs worse than GCN and even the worst among the benchmarks in some cases. The reason may be that the attention mechanism of GAT prefers to give more weights to neighbors with more deviated features, on the opposite to our low-pass 'message passing', which brings more risks to be attacked. This result implies that an attention model designed for better effectiveness on clean graphs does not necessarily work well under attacks.

Additionally, using our Greedy algorithm as the attacking model, we present the results on the Cora, Citeseer, and Pubmed datasets in Fig. 3. Our low-pass method still performs better than the benchmark ones. The gap is large especially when the number of perturbations is small. This is in accordance with the error bound obtained in Theorem 2. With fewer perturbations, i.e., a smaller Δ, the error bound is lower, which helps to protect the model from the graph structural attacks effectively. But with more perturbations, the prediction deviations for different nodes increase, which makes the model more sensitive to the graph structural changes.

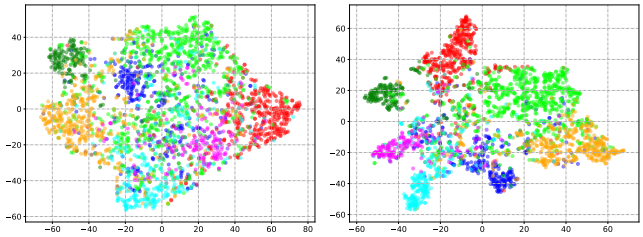
Indeed, we find that most algorithms perform similarly between evasion and poisoning attacks. This is because in either case, the adversary's main route of attack is to add



(a) Evasion on `Cora` (b) Evasion on `Citeseer` (c) Evasion on `Pubmed`

(d) Poisoning on `Cora` (e) Poisoning on `Citeseer` (f) Poisoning on `Pubmed`

Figure 3: Classification accuracy vs. number of perturbations per node. Results of node classification with the random splits of 20 labeled examples per class are presented. We report the mean values of 2000 trials on 20 splits with randomly selected targeted nodes. We apply the Greedy method to attack the graph structures.



(a) GCN (b) Low-Pass GCN

Figure 4: Low-Pass 'message passing' helps GCNs to learn more robust representations. We visualize the final-layer hidden representations of the all the nodes in the Cora dataset given by GCN and the GCN with low-pass 'message passing' by t-SNE [28]. We apply the random attacks on all the nodes, with $\Delta = 0.5$.

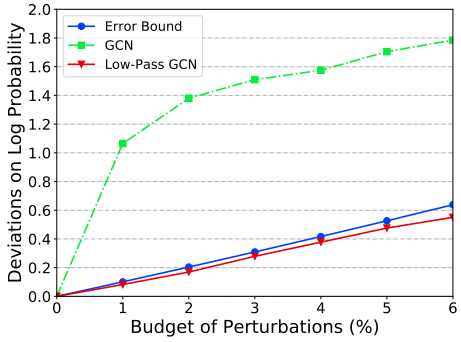


Figure 5: We provide an effective theoretical bound with our low-pass 'message passing'. We plot largest element-wise deviations vs. adversarial attack budget Δ. The blue line is the theoretical upper bound on the deviation of log probabilities derived for our proposed method.

edges from a node to other nodes belonging to different classes, to attempt to cause misclassifications.

C. Random Attacks and Theoretical Guarantees

In this section, we evaluate the effectiveness of the proposed methods under random attacks and observe whether the proposed theoretical guarantees are reliable. Given the budget Δ, we apply the globally random attacks applied on all the nodes (not only in the test set), instead of only attacking a small ratio of nodes. This could be more challenging and degrade the classification performance among all the nodes heavily.

Firstly, we perform the random evasion attacks on the Cora dataset over all the nodes with the attacking budgets being $\Delta = 0.5$. We visualize the final-layer hidden represen-

Table III: Node classification accuracy (%) of our low-pass 'message passing' with different R values. We report mean values of 2000 trials on 20 splits attacked by RL-S2V. The best value at each row is **boldface**.

Dataset	0.1	0.2	0.5	1	2	5	10
Cora	58.1	63.8	66.1	**66.2**	65.1	53.2	51.4
Citeseer	52.1	56.3	62.1	**62.3**	58.3	42.9	42.6
Pubmed	71.2	72.1	**75.3**	75.1	72.1	62.2	60.6
Co-CS	71.6	72.3	76.8	**77.1**	75.1	63.2	62.9
Co-Phy	72.1	73.5	79.1	**79.2**	77.4	63.5	63.2

tations given by GCN and the GCN with low pass 'message passing' presented in Figure 4. It is obvious that, under the attacks of the same budget, the hidden representations of different classes given by the low-pass method form tighter clusters than GCN, i.e., the representations belonging to the same class are concentrated more. This demonstrates that our low-pass message passing method helps to achieve better robustness on learning semantic features for nodes, and potentially improves the node classification accuracy under attacks.

Furthermore, we conduct the experiments to see whether the derived bound is reliable. We randomly generate the graphs with $|\mathcal{V}| = 10000$ and the average degree being 100 [35], and the sparse binary features of 1000 dimensions with 20 non-zero elements per node. The weights are generated by standard Gaussian variables and normalized row-wise so that $\|\mathbf{W}^{(l)}\| = 1, \forall l$ holds. We perform random attacks to the graph, which insert or delete edges between randomly selected nodes the graph, as introduced in [21]. Figure 5 plots the largest deviations in log probability $|\log \check{P}_{ic} - \log P_{ic}|$ for any node i and any class c caused by the random attacks. The error bound for the GCN with our low-pass 'message passing' is obtained by Theorem 2. It can be seen that large performance gaps exist between our approach and GCN on the deviations since the proposed low-pass aggregation matrix prevents excessive perturbations arising from the graph structural attacks. Besides, the derived error bound is reliable even when faced with random attacks applied on the whole graph.

D. Experiments on Parameter Sensitivity

In this section, we present the results of the experiments on the effects of the parameter selections on the performance of our low-pass 'message passing' approach . We conduct the experiments with different R values under the NL-S2V evasion attacks. The experimental setting is as same as that for Table II. We report the result in Table III. We can see that the model is more robust with a smaller R against adversarial attacks. The reason for this is that a low R ensures that any neighbor of a node cannot have large effects through 'message passing' so that no matter how strong the attack is, the classification results rely on the input features of each node but are not greatly affected by the graph structural changes. Also, the proposed Theorem 1 and Theorem 2 demonstrate this point, i.e., a small value R induces a low error bound. For larger R values, the model is less robust. This is because large R cannot defend against attacks from camouflage edges effectively. Overall, the performance is not sensitive to the value of R between 0.5 and 2. Thus, empirically, we choose $R = 1$ as the default setting in all of our experiments and show that we achieve satisfactory performance with it.

VI. CONCLUSION

In this work, we propose the low-pass 'message passing', a novel mechanism against adversarial edge additions and deletions. The core design is a low-pass aggregation matrix, which modifies the propagation strength of different edges based on their reliability. We validate the effectiveness of our method on defending perturbations through comprehensive experiments and additionally provide provable robustness guarantees against the graph structural attacks for the task of node classification.

An interesting future direction is to extend our low-pass 'message passing' mechanism to other tasks on the graph data, such as link prediction and graph classification. In addition, since the selection of the hyper-parameters of our approach can influence the robustness of different GCN models, how to use the approaches, e.g., Bayesian Optimization, to automatically select the optimal values for them is worth exploring.

ACKNOWLEDGMENT

This paper is supported by NUS ODPRT Grant R252-000-A81-133, Singapore Ministry of Education Academic Research Fund Tier 3 under MOEs official grant number MOE2017-T3-1-007, and NFS of China, No. 61772082.

REFERENCES

[1] B. London and L. Getoor, "Collective classification of network data." *Data Classification: Algorithms and Applications*, vol. 399, 2014.

[2] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2016.

[4] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Column networks for collective classification," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[6] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2847–2856.

[7] A. Tamersoy, "Graph-based algorithms and models for security, healthcare, and finance," Ph.D. dissertation, Georgia Institute of Technology, 2016.

[8] F. Wu, T. Zhang, A. Holanda de Souza, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *Proceedings of Machine Learning Research*, 2019.

[9] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[10] N. Weaver, *Lipschitz algebras*. World Scientific, 1999.

[11] L. Maligranda *et al.*, "Some remarks on the triangle inequality for norms," *Banach Journal of Mathematical Analysis*, vol. 2, no. 2, pp. 31–41, 2008.

[12] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," *arXiv preprint arXiv:1806.03536*, 2018.

[13] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations. https://openreview. net/forum*, 2020.

[14] G. Li, M. Müller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" *arXiv preprint arXiv:1904.03751*, 2019.

[15] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *arXiv preprint arXiv:1811.05868*, 2018.

[16] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *International Conference on Machine Learning*, 2018, pp. 1123–1132.

[17] Z. Tong, Y. Liang, C. Sun, D. S. Rosenblum, and A. Lim, "Directed graph convolutional network," *arXiv preprint arXiv:2004.13970*, 2020.

[18] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2272–2281.

[19] Y. Wang, W. Wang, Y. Liang, Y. Cai, J. Liu, and B. Hooi, "Nodeaug: Semi-supervised node classification with data augmentation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 207–217.

[20] Y. Wang, W. Wang, Y. Liang, Y. Cai, and B. Hooi, "Graphcrop: Subgraph cropping for graph classification," 2020.

[21] D. Zhu, Z. Zhang, P. Cui, and W. Zhu, "Robust graph convolutional networks against adversarial attacks," 2019.

[22] L. Sun, J. Wang, P. S. Yu, and B. Li, "Adversarial attack and defense on graph data: A survey," *arXiv preprint arXiv:1812.10528*, 2018.

[23] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *International Conference on Learning Representations (ICLR)*, 2019.

[24] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *arXiv preprint arXiv:1902.08226*, 2019.

[25] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu, "Adversarial examples for graph data: Deep insights into attack and defense," in *International Joint Conference on Artificial Intelligence, IJCAI*, 2019, pp. 4816–4823.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[27] X. Wang, J. Eaton, C.-J. Hsieh, and F. Wu, "Attack graph convolutional networks by adding fake nodes," *arXiv preprint arXiv:1810.10751*, 2018.

[28] A. Buja, D. Cook, and D. F. Swayne, "Interactive high-dimensional data visualization," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 78–99, 1996.

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.

[30] X. Tang, Y. Li, Y. Sun, H. Yao, P. Mitra, and S. Wang, "Transferring robustness for graph neural network against poisoning attacks," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 600–608.

[31] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," *arXiv preprint arXiv:2005.10203*, 2020.

[32] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[34] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 895–904.

[35] V. Batagelj and U. Brandes, "Efficient generation of large random networks," *Physical Review E*, vol. 71, no. 3, p. 036113, 2005.