Adversarial Immunization for Improving Certifiable Robustness on Graphs

Shuchang Tao[12], Huawei Shen[1], Qi Cao[12], Liang Hou[12], Xueqi Cheng[1]
[1]CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Despite achieving strong performance in the semi-supervised node classification task, graph neural networks (GNNs) are vulnerable to adversarial attacks, similar to other deep learning models. Existing research works either focus on developing robust GNN models or attack detection methods against attacks on graphs. However, little research attention is paid to the potential and practice of immunization to adversarial attacks on graphs. In this paper, we formulate the problem of *graph adversarial immunization* as a bilevel optimization problem, i.e., vaccinating an affordable fraction of node pairs, connected or unconnected, to improve the certifiable robustness of the graph against any admissible adversarial attack. We further propose an efficient algorithm, called *AdvImmune*, which optimizes meta-gradient in a discrete way to circumvent the computationally expensive combinatorial optimization when solving the adversarial immunization problem. Experiments are conducted on two citation networks and one social network. Experimental results demonstrate that the proposed *AdvImmune* immunization method remarkably improves the fraction of robust nodes by 12%, 42%, 65%, with an affordable immune budget of only 5% edges.

CCS CONCEPTS

• **Networks** → **Network reliability**; • **Theory of computation** → **Machine learning theory**.

KEYWORDS

Adversarial immunization; Graph neural networks; Node classification; Certifiable robustness.

ACM Reference Format:
Shuchang Tao[12], Huawei Shen[1], Qi Cao[12], Liang Hou[12], Xueqi Cheng[1]. 2021. Adversarial Immunization for Improving Certifiable Robustness on Graphs . In *Proceedings of WSDM '21: 14th ACM International WSDM Conference (WSDM '21).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Graph data are ubiquitous in the real world, characterizing complex relationships among objects or entities. Typical graph data include social networks, citation networks, and traffic networks. In the last few years, graph neural networks (GNNs) emerge as a family of powerful tools to model graph data, achieving remarkable performance in many tasks of graphs such as node classification [20, 21, 35, 36], link prediction [10], and cascade prediction [4, 24]. Despite their success, similar to other deep learning models [5], GNNs are proved to be vulnerable to adversarial attacks [7, 42], i.e., imperceptible perturbations on graph structure or node feature can easily fool GNNs. This poses a fundamental challenge for protecting GNNs from adversarial attacks on graph data.

Many researchers devote to designing defense methods against adversarial attacks to GNN models, including adversarial training [8, 11] and attack detection [39]. Defense methods spring up rapidly and gain success at improving the performance of GNN models [34, 37, 41]. However, these methods are usually heuristic [2] and only effective for certain attacks rather than all attacks. Consequently, an endless cat-and-mouse game or competition emerges between adversarial attacks and defense methods [2]. Recently, to solve the attack-defense dilemma, researchers resort to robustness certification [22, 45] and robust training on graphs against any admissible adversarial attack [2, 43]. However, such robust training may damage the performance of GNN models on clean graph, which is undesirable before the adversarial attack actually happens.

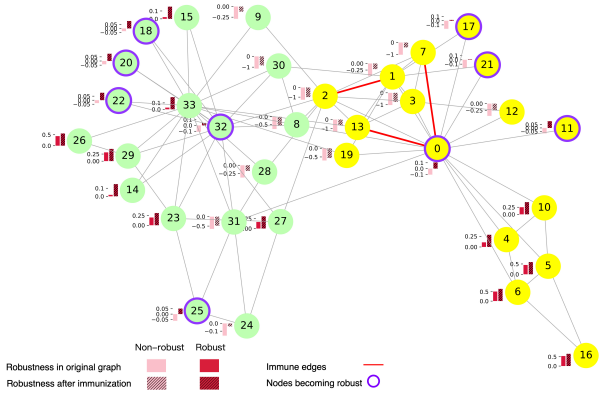


Figure 1: Effect of *AdvImmune* immunization on Karate club network. For each node, we use two bars to represent its robustness before and after immunization. When the value of robustness is larger than 0, the corresponding node is certified as robust (red bar), otherwise as non-robust (pink bar). Purple circle indicates the node that becomes robust after immunization. The red edges are the immune edges.

WSDM '21, March 8–12, 2021, Jerusalem, Israel

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Up to now, little research attention is paid to the potential and practice of immunization to adversarial attacks on graphs. In this paper, we firstly propose and formulate *adversarial immunization*, which is the first action guideline to improve the certifiable robustness against any admissible adversarial attack from the perspective of graph structure instead of GNN models or training methods. Specifically, *adversarial immunization vaccinates a fraction of node pairs in advance, connected or unconnected, to protect them from being modified by attacks*, making the whole graph more robust to adversarial attacks. Note that, adversarial immunization is general and flexible, not only improving the certifiable robustness against any admissible attack, but also avoiding the cons of performance drop on unattacked clean graph suffered by robust training [2]. We further propose an efficient algorithm called *AdvImmune* to obtain the target immune node pairs with meta-gradient in a discrete way, circumventing the computationally expensive combinatorial optimization when solving the adversarial immunization problem.

To offer an intuitive understanding about the effect of *AdvImmune* immunization on improving node robustness, we illustrate *AdvImmune* on an example network, i.e., the Karate club network. As shown in Fig. 1, only immunizing 3 edges in advance, the number of nodes certified as robust against any admissible attack increases by 9. Indeed, the remarkable improvement of certifiable robustness is also observed when applying our proposed *AdvImmune* method on real large-scale networks. Experimental results on Reddit show that with an affordable immune budget of only 5% edges, the fraction of robust nodes improves by 65%. Such results indicate the effectiveness of our proposed *AdvImmune* method, i.e., immunizing certain affordable critical node pairs in advance can significantly improve the certifiable robustness of the graph.

Adversarial immunization also has much potential in real world application. For example, in a recommendation system, immunization protects users from junk or low-quality products by maintaining the relationship between real users and products; in a credit scoring [18] system, immunization can maintain certain critical relation of user pairs to prevent fraudsters from pretending to be customers, avoiding serious financial losses.

In summary, our contributions are as follows:

(1) To the best of our knowledge, this is the first work proposing and formalizing *adversarial immunization* on graphs to improve certifiable robustness against any admissible attack, providing a new insight into graph adversarial learning.
(2) We propose *AdvImmune* to innovatively tackle adversarial immunization with meta-gradient to obtain the optimal immune node pairs, thus circumventing the challenges in computationally expensive combinatorial optimization.
(3) We demonstrate the effectiveness of our *AdvImmune* method on both citation network and social network datasets. Experimental results show that our proposed method significantly improves the fraction of robust nodes by 12%,42%,65%, with an affordable immune budget of only 5% edges.

2 PRELIMINARIES

Since the adversarial learning on graphs mainly take the semi-supervised node classification as the target task, this section first introduces the task of semi-supervised node classification, together with a widely used GNN model to tackle this task. Besides, we also introduce the robustness certification against any admissible attack.

Semi-supervised node classification. Given an attributed graph $G = (\boldsymbol{A}, \boldsymbol{X})$, $\boldsymbol{A} \in \{0,1\}^{N\times N}$ is the adjacency matrix and $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ is the attribute matrix consisting of node attributes, N is the number of nodes and d is the dimension of node attributes. We denote the node set as $\mathcal{V} = \{1, 2, ..., N\}$ and the edge set as $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. In the semi-supervised node classification, a subset of nodes $\mathcal{V}_l \in \mathcal{V}$ are labelled from class sets $\mathcal{K}$. The goal is to assign a label for each unlabelled node by the learned classifier $\boldsymbol{Y} = f(\boldsymbol{A}, \boldsymbol{X}) \in \mathbb{R}^{N\times K}$, where $K = |\mathcal{K}|$ is the number of classes.

Graph neural networks. GNNs have achieved a remarkable success on the semi-supervised node classification task [15, 20, 21, 31]. Among existing GNN models, π-PPNP [21] shows an outstanding performance. It connects graph convolutional networks (GCN) to PageRank to effectively capture the impact of infinitely neighborhood aggregation, and decouples the feature transformation from propagation to simplify model structure. In this paper, we consider π-PPNP as the representative of GNN models to tackle node classification problem. The formulation is as following:

$$\begin{aligned} \boldsymbol{H} &= f_\theta(\boldsymbol{X}) \\ \boldsymbol{Y} &= \operatorname{softmax}(\boldsymbol{\Pi}\boldsymbol{H}), \end{aligned} \tag{1}$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times K}$ is the transformed features computed by a neural network f_θ, $\boldsymbol{H}^{\text{diff}} := \boldsymbol{\Pi}\boldsymbol{H}$ is defined as the *diffused logits*. Note that, the diffused logits are the unnormalised outputs as real numbers ranging from $(-\infty, +\infty)$, which are also referred to raw predictions. $\boldsymbol{\Pi} = (1-\alpha)\left(\boldsymbol{I}_N - \alpha\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{-1}$ is personalized PageRank [23] that measures distance between nodes, and $\boldsymbol{D}$ is a diagonal degree matrix with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$. The personalized PageRank with source node t on graph G is written as:

$$\boldsymbol{\pi}_G(\boldsymbol{e}_t) = (1-\alpha)\boldsymbol{e}_t\left(\boldsymbol{I}_N - \alpha\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{-1}, \tag{2}$$

where $\boldsymbol{e}_t$ is a the t-th canonical basis vector (row vector). $\boldsymbol{\pi}_G(\boldsymbol{e}_t) = \boldsymbol{\Pi}_{t,:}$ is the transportation of the $t-$th row of personalized PageRank matrix $\boldsymbol{\Pi}$.

Robustness certification. Since GNN models are vulnerable to adversarial attacks, Bojchevski *et al.* [2] certify the robustness of each node against any admissible attack on graph by *worst-case margin*. Specifically, the difference between the raw predictions of node t on label class y_t and other class k defines the *margin* on the perturbed graph $\tilde{G}$. Taking π-PPNP as a typical GNN model, the formula of the defined margin is as follow:

$$\mathbf{m}_{y_t,k}(t, \tilde{G}) = \boldsymbol{H}^{\text{diff}}_{t,y_t} - \boldsymbol{H}^{\text{diff}}_{t,k} = \boldsymbol{\pi}_{\tilde{G}}(\boldsymbol{e}_t)\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right), \tag{3}$$

where y_t is the label class of node t.

For a target node t, the *worst-case margin* between class y_t and class k under any admissible perturbation $\tilde{G} \in \boldsymbol{Q}_{\mathcal{F}}$ is:

$$\mathbf{m}_{y_t,k}(t, \tilde{G}^*) = \min_{\tilde{G}\in\boldsymbol{Q}_{\mathcal{F}}} \mathbf{m}_{y_t,k}(t, \tilde{G}) = \min_{\tilde{G}\in\boldsymbol{Q}_{\mathcal{F}}} \boldsymbol{\pi}_{\tilde{G}}(\boldsymbol{e}_t)\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right), \tag{4}$$

where $\tilde{G} = (\mathcal{V}, \tilde{\mathcal{E}} := \mathcal{E}_f \cup \mathcal{F}_+)$ is the perturbed graph consisting of fixed edges $\mathcal{E}_f$ and the perturbed edges $\mathcal{F}_+$, $\tilde{G}^*$ is the worst-case perturbed graph. The set of admissible perturbed graphs $\boldsymbol{Q}_{\mathcal{F}} = \left\{\left(\mathcal{V}, \tilde{\mathcal{E}} := \mathcal{E}_f \cup \mathcal{F}_+\right) | \mathcal{F}_+ \subseteq \mathcal{F}, \left|\tilde{\mathcal{E}} \backslash \mathcal{E}\right| + \left|\mathcal{E} \backslash \tilde{\mathcal{E}}\right| \leq B, \left|\tilde{\mathcal{E}}^t \backslash \mathcal{E}^t\right| +\right.$

$\left|\mathcal{E}^t \backslash \tilde{\mathcal{E}}^t\right| \leq b_t, \forall t\right\}$, where $\mathcal{F} \subseteq (\mathcal{V}\times\mathcal{V})\backslash\mathcal{E}_f$ is fragile edge set. Each fragile edge $(i, j) \in \mathcal{F}$ can be included in the graph or excluded from the graph by attacker, and the selected attack edge $\mathcal{F}_+$ is a subset of $\mathcal{F}$. The perturbations satisfy both global budget and local budget, where global budget $|\tilde{\mathcal{E}}\backslash\mathcal{E}| + |\mathcal{E}\backslash\tilde{\mathcal{E}}| \leq B$ requires that there are at most B perturbing edges, and local budget $|\tilde{\mathcal{E}}^t\backslash\mathcal{E}^t| + |\mathcal{E}^t\backslash\tilde{\mathcal{E}}^t| \leq b_t$ limits node t to have no more than b_t perturbing edges.

Node t is certifiably robust when

$$\mathbf{m}_{y_t,k_t}(t, \tilde{G}^*) = \min_{k\neq y_t} \mathbf{m}_{y_t,k}(t, \tilde{G}^*) > 0, \tag{5}$$

where k_t is the most likely class among other classes. In other words, whether a node is robust is determined by the worst-case margin against any admissible perturbed graph being greater than zero. This certification method can directly measure the robustness of the graph under GNN model.

Bojchevski *et al.* [2] use policy iteration with reward $\boldsymbol{r} = -\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right)$ to find the worst-case perturbed graph. Under certain set of admissible perturbed graph $\boldsymbol{Q}_{\mathcal{F}}$, running policy iteration $K\times(K-1)$ times can obtain the certificates for all N nodes.

3 ADVERSARIAL IMMUNIZATION

In this section, we first formalize the problem of adversarial immunization and elaborate it with a widely used GNN model. Then we propose an effective algorithm *AdvImmune*, using meta-gradient for selecting and immunizing appropriate node pairs in advance, to improve the certifiable robustness of the graph.

3.1 Problem Formulation

Adversarial immunization aims to improve the certifiable robustness of nodes against any admissible attack, i.e. the minimal *worst-case* margin of nodes under the node classification task. Specifically, by immunizing appropriate node pairs in advance, GNN model can correctly classify nodes even under the worst case. In this paper, we use π-PPNP as the typical GNN model, since it shows outstanding performance on node classification task. The general goal of adversarial immunization is formalized as:

$$\max_{\mathcal{E}_{\boldsymbol{c}}\in\mathcal{S}_{\boldsymbol{c}}} \min_{k\neq y_t} \mathbf{m}_{y_t,k}(t, \hat{G}) = \max_{\mathcal{E}_{\boldsymbol{c}}\in\mathcal{S}_{\boldsymbol{c}}} \min_{k\neq y_t} \boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t)\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right), \tag{6}$$

where $\boldsymbol{\pi}_{\hat{G}}$ is the personalized PageRank of the modified graph, $\hat{G} = \left(\mathcal{V}, \hat{\mathcal{E}} := \left(\mathcal{E}_{\tilde{G}^*} \cup \mathcal{E}_{\boldsymbol{c}}^{connect}\right)\backslash\mathcal{E}_{\boldsymbol{c}}^{unconnect}\right)$ is the modified graph with contribution of both perturbed graph $\mathcal{E}_{\tilde{G}^*}$ and immune graph $\mathcal{E}_{\boldsymbol{c}} = \left(\mathcal{E}_{\boldsymbol{c}}^{connect} \cup \mathcal{E}_{\boldsymbol{c}}^{unconnect}\right) \subseteq (\mathcal{V}\times\mathcal{V})$, $\mathcal{E}_{\tilde{G}^*}$ is the edge set of the worst-case perturbed graph $\tilde{G}^*$:

$$\tilde{G}^* = \arg\min_{\tilde{G}\in\boldsymbol{Q}_{\mathcal{F}}} \mathbf{m}_{y_t,k}(t, \tilde{G}), \tag{7}$$

and the immune graph $\mathcal{E}_{\boldsymbol{c}}$ contain both connected edges $\mathcal{E}_{\boldsymbol{c}}^{connect}$ to keep them in the graph and unconnected node pairs $\mathcal{E}_{\boldsymbol{c}}^{unconnect}$ to keep them not in the graph. Due to the limited immune budget in reality, we cannot immunize all node pairs. Here, we consider both local budget and global budget to constrain the choice of immune node pairs $\mathcal{E}_{\boldsymbol{c}}$. Globally, the number of immune node pairs should be no more than *global budget* C, i.e. $|\mathcal{E}_{\boldsymbol{c}}| \leq C$. For each node t, the number of immune node pairs cannot exceed the *local budget* c_t, i.e. $\left|\mathcal{E}_{\boldsymbol{c}}^t\right| \leq c_t$, where $\mathcal{E}_{\boldsymbol{c}}^t$ represents the node pairs associated with node t. The set of admissible immune node pairs $\mathcal{E}_{\boldsymbol{c}}$ is defined as:

$$\mathcal{S}_{\boldsymbol{c}} = \left\{\mathcal{E}_{\boldsymbol{c}} | \mathcal{E}_{\boldsymbol{c}} \subseteq (\mathcal{V}\times\mathcal{V}), |\mathcal{E}_{\boldsymbol{c}}| \leq C, \left|\mathcal{E}_{\boldsymbol{c}}^t\right| \leq c_t, \forall t \in \mathcal{V}\right\}. \tag{8}$$

Note that, the above immunization objective function is for a single node t. In order to improve the certifiable robustness of the entire graph, we take the sum of the worst-case margins over all nodes as the overall goal of adversarial immunization:

$$\max_{\mathcal{E}_{\boldsymbol{c}}\in\mathcal{S}_{\boldsymbol{c}}} \sum_{t\in\mathcal{V}} \min_{k\neq y_t} \boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t)\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right). \tag{9}$$

Challenges: It's not easy to obtain the optimal immune node pairs due to two issues. First, the computational cost on selecting the set of certain node pairs from the total node pairs is expensive. Given a global immune budget of C, the possible immunization plans is $\binom{N^2}{C}$. This leads to an unbearable search cost $O\left(N^{2C}\right)$, making it difficult to find the optimal immune node pairs efficiently. The second challenge comes from the discrete nature of graph data. The resulting non-differentiability hinders back-propagating gradients to guide the optimization of immune node pairs.

3.2 The Proposed Method for Adversarial Immunization

We propose *AdvImmune* to greedily obtain the solution via meta-gradient, addressing the adversarial immunization effectively. To facilitate the solution of meta-gradient, we first regard immune node pairs as hyperparameters in matrix form.

Matrix form of the problem. We use the adjacency matrix to represent the set of immune node pairs in Eq. 9 and formalize our goal with the matrix form:

$$\max_{\mathcal{A}_{\boldsymbol{c}}\in\mathcal{A}_{\mathcal{S}_{\boldsymbol{c}}}} \sum_{t\in\mathcal{V}} \min_{k\neq y_t} \boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t)\left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k}\right), \tag{10}$$

$$\boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t) = (1-\alpha)\boldsymbol{e}_t\left(\boldsymbol{I}_N - \alpha\boldsymbol{D}_{\hat{G}}^{-1}\boldsymbol{A}_{\hat{G}}\right)^{-1}, \quad \boldsymbol{A}_{\hat{G}} = \boldsymbol{A} + \boldsymbol{A}'_{\tilde{G}^*} * \mathcal{A}_{\boldsymbol{c}},$$

where $\mathcal{A}_{\boldsymbol{c}}$ is the matrix form of immune graph $\mathcal{E}_{\boldsymbol{c}}$, $\boldsymbol{A}'_{\tilde{G}^*}$ is the matrix form of the perturbing edge set corresponding to the worst-case perturbed graph $\tilde{G}^*$, $\boldsymbol{A}_{\hat{G}}$ is the adjacency matrix of the modified graph $\hat{G}$ with the contribution of both worst-case perturbing graph $\boldsymbol{A}'_{\tilde{G}^*}$ and immune graph $\mathcal{A}_{\boldsymbol{c}}$, and $*$ is element-wise multiplication. In immune graph $\mathcal{A}_{\boldsymbol{c}}$, 0 indicates that the corresponding node pair will be immunized, which will filter the effect of perturbations, while 1 implies that the corresponding node pair is not immunized and may be attacked. In other words, $\mathcal{A}_{\boldsymbol{c}}$ can be regarded as a mask, protecting these immune node pairs from being modified or attacked. Such a matrix form transforms the original computationally expensive combinatorial optimization of node pair set into a discrete matrix optimization problem, reducing the difficulty of obtaining the optimal solution.

The next and key problem is how to solve the optimization problem with discrete matrix form. We innovatively tackle this problem by greedy algorithm via meta-gradient to obtain the optimal immune graph matrix.

Meta-gradient of immune graph matrix. Meta-gradient is referred as the gradient of hyperparameters [12, 43]. Regarding the

Algorithm 1 *AdvImmune* immunization on graphs

Input: Graph $G = (\boldsymbol{A}, \boldsymbol{X})$, *immune budget* (C, c),
Output: Immune graph matrix $\mathcal{A}_{\boldsymbol{c}}$
$\tilde{G}^* \leftarrow$ train surrogate model to get worst-case graph
Initialize the immune graph matrix as $\mathcal{A}_{\boldsymbol{c}} = ones((N, N))$
while number of immune node pairs in $\mathcal{A}_{\boldsymbol{c}} < C$ **do**
  $k_t \leftarrow$ the class which minimize the worst-case margin of each node t as Eq. 12
  $\hat{G} \leftarrow$ after perturbing with $\tilde{G}^*$ of the worst-case class k_t and immunizing with $\mathcal{A}_{\boldsymbol{c}}$.
  $\nabla^{\text{meta}}_{\mathcal{A}_{\boldsymbol{c}}} \leftarrow$ the meta-gradient computing in Eq. 11
  Value $V_{(i,j)} \leftarrow -\nabla^{\text{meta}}_{a_{(i,j)}}$
  Select the node pairs that have already been immunized and set the corresponding entries to 0 in V.
  $(i^*, j^*) \leftarrow$ maximal entry in V which satisfy c.
  $\mathcal{A}_{\boldsymbol{c}}[i^*, j^*] = 0 \leftarrow$ immunize one more node pair (i^*, j^*).
end
Return: $\mathcal{A}_{\boldsymbol{c}}$

immune graph matrix $\mathcal{A}_{\boldsymbol{c}}$ as a hyperparameter, we can calculate the meta-gradient of the matrix $\mathcal{A}_{\boldsymbol{c}}$ with the objective function:

$$\nabla^{\text{meta}}_{\mathcal{A}_{\boldsymbol{c}}} = \nabla_{\mathcal{A}_{\boldsymbol{c}}} \left[\sum_{t \in \mathcal{V}} \min_{k \neq y_t} \boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t) \left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k} \right) \right], \tag{11}$$

where $\nabla^{\text{meta}}_{\mathcal{A}_{\boldsymbol{c}}}$ is the meta-gradient of the immune graph matrix $\mathcal{A}_{\boldsymbol{c}}$. Each entry in $\nabla^{\text{meta}}_{\mathcal{A}_{\boldsymbol{c}}}$ represents the effect of the corresponding node pair on the objective function, i.e., worst-case margin. Note that, before computing the meta-gradient, we calculate the most likely class k_t which minimizes the *worst-case* margin of each node t.

$$k_t = \arg\min_{k \neq y_t} \boldsymbol{\pi}_{\hat{G}}(\boldsymbol{e}_t) \left(\boldsymbol{H}_{:,y_t} - \boldsymbol{H}_{:,k} \right). \tag{12}$$

Directly performing gradient optimization may result in decimals and the elements greater than 1 or smaller than 0 in the matrix, making $\mathcal{A}_{\boldsymbol{c}}$ no longer a matrix indicating the immune choice of node pairs in graph. To solve this challenge, we optimize the immune graph matrix in a discrete way by greedy algorithm.

Greedy immunization via meta-gradient. Since the objective in immunization is a maximization problem, we apply discrete gradient ascent to solve it. In other words, we calculate the meta-gradient for each entry of the matrix $\mathcal{A}_{\boldsymbol{c}}$, and choose the node pair with greatest effect greedily. Since $\mathcal{A}_{\boldsymbol{c}}$ is initialized as a matrix with all elements of 1, it can only be changed from 1 to 0. Hence, only the negative gradient is helpful. We define the inverse of the meta-gradient as the value of corresponding node pair: $V_{(i,j)} = -\nabla^{\text{meta}}_{\mathcal{A}_{\boldsymbol{c}_{(i,j)}}}$, which represents the impact of the corresponding node pair on the goal of adversarial immunization.

In order to preserve the node pairs with greatest influence, we greedily select the entry with maximum value in V:

$$(i^*, j^*) = \arg\max_{(i,j)} V_{(i,j)}, \tag{13}$$

and set the corresponding entries in $\mathcal{A}_{\boldsymbol{c}}$ to zero as immune node pairs, protecting them from being attacked and modified.

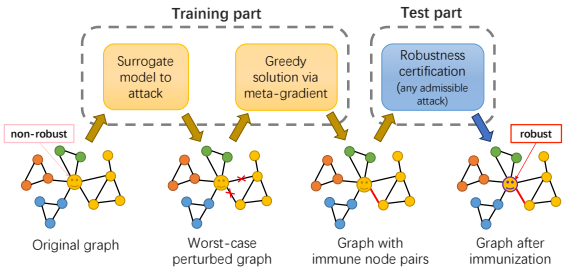


Figure 2: The training and test process of *AdvImmune*.

3.3 Algorithm

In this section, we describe the *AdvImmune* algorithm to obtain immune node pairs to improve the certifiable robustness of the graph. We first initialize $\mathcal{A}_{\boldsymbol{c}}$ as a matrix with all elements of 1, indicating that no node pair to be immunized. Then we iteratively select the node pair with the greatest impact. In each iteration, we choose the most likely class k_t which minimize the worst-case margin of node t. Then the core step is to calculate the meta-gradient of the objective function for $\mathcal{A}_{\boldsymbol{c}}$ and the corresponding value $V_{(i,j)}$. We select the maximum entry in $V_{(i,j)}$ which satisfy *local budget* c, and set the corresponding entry in $\mathcal{A}_{\boldsymbol{c}}$ to zero. This process is repeated until we obtain enough immune node pairs.

We illustrate both the training and test process of adversarial immunization in Fig. 2. During the training process, we first use surrogate model to obtain the worst-case perturbed graph. Then, we select and immunize appropriate node pairs through Alg. 1 via meta-gradient. As for test (certificate), we use the immune graph as a mask to protect certain node pairs from being attacked, and improve the certifiable robustness of nodes on the graph against any admissible adversarial attack.

3.4 Complexity analysis

The computational complexity of *AdvImmune* immunization method depends on the GNN model. Suppose GNN has a computational complexity of $O(T)$, then the computational complexity of the surrogate attack model is also $O(T)$. Specifically, under GNN model of π−PPNP, the personalized PageRank matrix $\boldsymbol{\pi}$ is dense, leading to a computational complexity and memory requirement of $O(T) = O(N^2)$. Since we have to calculate the minimal worst-case margins for each pair of classes (k_1, k_2), the computational complexity in each iteration is $O\left(T \cdot K^2\right)$. As for the optimization process of immunization, operations of both element-wise multiplication and meta-gradient require a computational complexity and memory of $O(E)$. Considering the total of C iterations to find the optimal C node pairs to be immunized, immunization method has a computational complexity of $O\left(C \cdot K^2 \cdot (T + E)\right)$.

4 EXPERIMENTS

4.1 Dataset

We evaluate our proposed method on two commonly used citation networks: Citeseer and Cora-ML [2], and a social network Reddit [38]. In citation networks, a node represents a paper with key words as attributes and paper class as label, and the edge means the

Table 1: Statistics of the evaluation datasets

Dataset	Type	N_{LCC}	$\lvert\mathcal{E}_{LCC}\rvert$	d	K
Citeseer	Citation network	2110	3668	3703	6
Cora-ML	Citation network	2810	7981	2879	7
Reddit	Social network	3069	7009	602	5

citation relationship. In Reddit, each node represents a post with word vectors as attributes and community as label, while each edge means the post-to-post relationship.

Due to the high complexity of PPNP, it is difficult to apply on large graphs, leading to the limitation of our method. Therefore, we only keep a subgraph of Reddit to conduct the experiments. Specifically, we first randomly select 10,000 nodes, and four classes with the most nodes are selected as our target classes. All nodes in these target classes are kept. Then, in order to retain the network structure as much as possible, we further include the first-order neighbors of the kept nodes. Class besides the target four classes is marked as the fifth class, i.e., other-class.

Experiments are conducted on the largest connected component for both citation networks and social networks. The statistics of each dataset are summarized in Table 1.

4.2 Baseline

To the best of our knowledge, we are the first to propose adversarial immunization on graphs. To demonstrate the effectiveness of our *AdvImmune* immunization method, we design two random immunization methods, as well as several heuristic immunization methods as our baselines. Besides, we also compare our *AdvImmune* immunization method with the state-of-the art defense method.

• **Random immunization methods.** We consider two random methods with different candidate sets of immune node pairs. *Random* method selects immune node pairs randomly from all node pairs, while *Attack Random* selects immune node pairs from the worst-case perturbed edges obtained by the surrogate model. Note that, since *Attack Random* knows the worst-case perturbed edges of the surrogate model attacks, it is stronger than the *Random* baseline.

• **Heuristic immunization methods.** *(1) Attribute-based methods.* Researches show that attackers tend to remove edges between nodes from the same class and add edges to node pair from different classes [34, 43]. Therefore, we design baselines that maintain the node pairs with high similarity between the attributes of nodes under the same class and the disconnection of node pair with low attribute similarity under different classes. We consider the commonly used Jaccard score and cosine score as the measure of similarity, namely *Jaccard* and *Cosine* respectively. *(2) Structure-based methods.* We also design baselines considering the structure importance of edges. Specifically, for global structure importance, we use edge-betweenness as a measurement [13]. Locally, we adopt the Jaccard similarity between the neighbors' labels of node pair. These two indicators reflect the connectivity and importance of edges, and we heuristically choose the immune node pairs with greater values, namely *Betweenness* and *Bridgeness* respectively.

• **Defense methods.** We use the state-of-the-art defense method, robust training [2], as our strong baseline. Robust training aims to improve the certifiable robustness of the graph by retraining the GNN model for optimizing the *robust cross-entropy loss* and *robust hinge loss*, namely *RCE* and *CEM*. Note that, this is a strong baseline since it also devote to improving the certifiable robustness against any admissible attack. However, such robust training method suffers from the performance drop on clean graph and lacks flexibility to controlling the magnitude and effectiveness of defense.

4.3 Experimental Setup

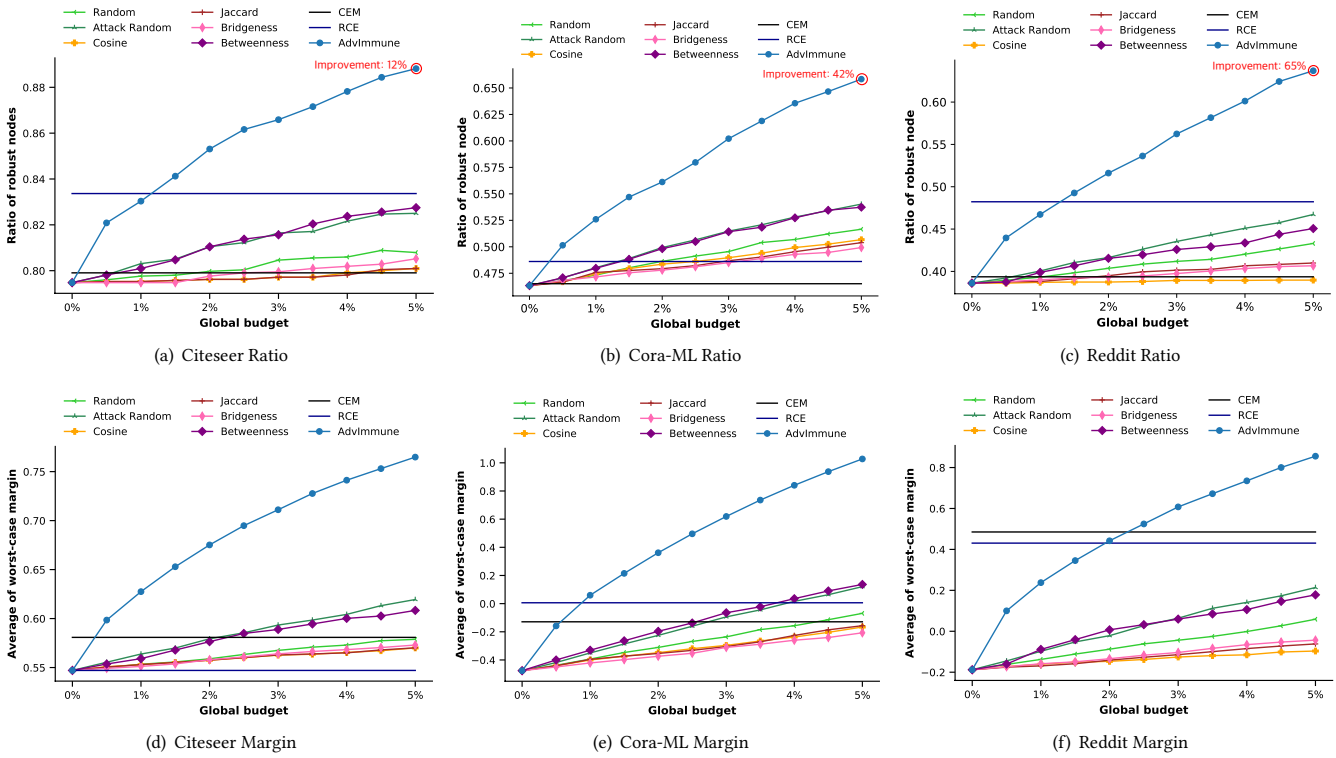
The settings of GNN model in our experiments, i.e., π-PPNP, are the same with [2]. Specifically, we set the transition probability $\alpha = 0.85$. As for our surrogate attack model, it has two settings of scenarios [2]. One is named as **Remove-only**, where $\mathcal{F} = \mathcal{E} \backslash \mathcal{E}_f$ for a given graph $G = (\mathcal{V}, \mathcal{E})$, i.e. attackers can only remove connected edges in the graph. The other is named as **Remove-Add**, where the fragile edge set is $\mathcal{F} = (\mathcal{V} \times \mathcal{V}) \backslash \mathcal{E}_f$, i.e. the attacker is allowed to add or remove connections of any node pair in the graph. Following the settings in robustness certification [2], the fixed edge set is $\mathcal{E}_f = \mathcal{E}_{mst}$, where $\mathcal{E}_{mst}$ is the edge of the minimum spanning tree (MST) of the graph, and the fragile edge is regarded as directed. For both the surrogate model and robustness certification, the local budget is limited as $b_t = \boldsymbol{D}_t$ in **Remove-only** scenario and $b_t = \max(\boldsymbol{D}_t - 6, 0)$ in **Remove-Add** scenario where $\boldsymbol{D}_t$ is the degree of node t. The global budget is $B = N^2$ in both scenarios, in order to better demonstrate the effectiveness of our proposed *AdvImmune* immunization even under strong attacks. Following the settings of robustness certification [2, 44], we also use the prediction class to measure the robustness of node. Besides, policy iteration is adopted to generate the worst-case perturbation.

For *AdvImmune* immunization, the immune *local budget* is $c_t = \boldsymbol{D}_t$ in **Remove-only** and is not set in **Remove-Add** scenario. For baselines, random-based methods randomize 10 times in both scenarios. As for attribute-based methods, in **Remove-only** scenario, they only immunize the connected edges with high attribute similarity between nodes under the same class, while in **Remove-Add** scenario, they immunize connected edges with a probability of 0.3 and unconnected node pairs with a probability of 0.7. Such settings of probability is based on the fact that there are about 30% removed edges among the worst-case perturbed edges in all datasets.

4.4 Effect of adversarial immunization

To comprehensively demonstrate the effectiveness of our proposed *AdvImmune* immunization, we conduct experiments on three datasets, i.e., Citeseer, Cora-ML, Reddit, and two certification scenarios, i.e., **Remove-only** and **Remove-Add**.

*4.4.1 Scenario of **Remove-only**.* In **Remove-only** scenario, the attacker can only remove the connected edges. We show the performance from two aspects: the number of robust nodes and the average of worst-case margins. In Fig. 3, the upper figures show the changes in the ratio of robust nodes when varying global budget of immunization from 0.5% to 5% edges. and the lower figures show the changes in the average of worst-case margins. Since only the node with positive worst-case margin can be certified as robust, the sudden changes of certifiable robustness resulting in sudden jumps in the ratio of robust nodes. As shown in Fig. 3, we can see that all immunization methods work better when global immune



(a) Citeseer Ratio (b) Cora-ML Ratio (c) Reddit Ratio

(d) Citeseer Margin (e) Cora-ML Margin (f) Reddit Margin

Figure 3: Immunization performance in the scenario of Remove-only. The upper figures (a-c) show the changes of the ratio of robust nodes, while lower figures (b-d) show the changes of the average of worst-case margins.

budget is increased, indicating that the more sufficient the immune budget is, the higher certifiable robustness of the graph is.

For random-based immunization methods, *Attack Random* performs better than *Random* on all datasets. This is because *Attack Random* knows the worst-case perturbed edges that the surrogate model attacks. However, even with the same access of the worst-case perturbed graph, *Attack Random* is far worse than our proposed *AdvImmune* method, further proving the superiority of our method.

For heuristic-based immunization methods, both *Jaccard* and *Cosine* methods even perform worse than *Random* method on three datasets. This result indicates that edges with high attribute similarity have less effect on certifiable robustness against any admissible attack, and may even bring some negative disturbance on the selection of immune node pairs. *Betweenness* is superior to *Bridgeness*, indicating that global structure importance is more effective than local structure importance when selecting immune node pairs.

For the state-of-the-art defense methods, *RCE* performs better than *CEM* on all datasets, which is consistent with the results in [2]. However, *AdvImmune* outperforms *RCE* when only immunizing 0.5% edges on Cora-ML, and immunizing 1.5% edges on Citeseer and Reddit. Our proposed *AdvImmune* method is very flexible, comparing to the defense models that are fixed after training.

As for our proposed *AdvImmune* immunization method, it significantly outperforms all the baselines on all datasets. On Reddit (Fig. 3(c), 3(f)), our *AdvImmune* method increases the ratio of robust nodes by 65% when immunizing only 5% of all edges. On the other two datasets, Citeseer and Cora-ML, our method also brings 12% and 42% improvement, when immunizing 5% edges. These results demonstrate the effectiveness of our proposed *AdvImmune* immunization method, i.e., significantly improves the certifiable robustness of the graph with an affordable immune budget.

4.4.2 Scenario of ***Remove-Add****.* We also conduct experiments in the scenario of **Remove-Add**, where the attacker is allowed to remove as well as add edges. In this scenario, immune node pairs contain both connected edges and unconnected node pairs, and the global budget is set to be 1% of all node pairs. Note that, the immunization in **Remove-Add** scenario is much difficult since the global budget of attack is $B = N^2$, which can thus better demonstrate the effectiveness of our proposed method. The baselines of structure-based immunization methods are ignored in this scenario since they can only calculate the importance of connected edges.

Tab. 2 shows the ratio of robust nodes and the improvement percentage after immunization or defense. The performance of immunization baselines are similar to that in the other scenario. *Attack Random* performs better than *Random*, while attribute-based heuristic immunization methods shows less effectiveness. As for defense baselines, *RCE* and *CEM*, they almost reach and perform the best among other heuristic immunization baselines.

As for our proposed *AdvImmune* method, it significantly outperforms all the baselines on all datasets, achieving the improvement of robust nodes by 32%, 46%,102%. Especially on Cora-ML, our

Table 2: Immunization performance in the scenario of Remove-Add

Dataset	Methods	Ratio of robust nodes	Improvement
Citeseer	No defense	0.4806	0
	Random	0.4807	0.02%
	Attack Random	0.5569	15.88%
	Jaccard	0.4910	2.17%
	Cosine	0.4910	2.17%
	CEM	0.5246	9.17%
	RCE	0.5441	13.21%
	AdvImmune	**0.6336**	**31.85%**
Cora-ML	No defense	0.1306	0
	Random	0.1310	0.27%
	Attack Random	0.1867	42.92%
	Jaccard	0.1552	18.80%
	Cosine	0.1577	20.71%
	CEM	0.1598	22.34%
	RCE	0.1335	2.18%
	AdvImmune	**0.2641**	**102.18%**
Reddit	No defense	0.2727	0
	Random	0.2729	0.06%
	Attack Random	0.3486	27.82%
	Jaccard	0.2822	3.46%
	Cosine	0.2822	3.46%
	CEM	0.3747	35.76%
	RCE	0.3832	40.50%
	AdvImmune	**0.3982**	**46.00%**

immunization method doubles the number of robust nodes. On Citeseer and Reddit, our method also improves the number of robust nodes by almost half. These results further demonstrate the effectiveness of our proposed *AdvImmune* immunization method even in a difficult scenario.

4.5 Transferability of immunization under various attacks

To verify the generality and transferability of the immune node pairs obtained by our proposed *AdvImmune* method, we evaluate its effectiveness against various attacks, i.e., state-of-the-art *metattack* [43] and *surrogate attack* [2] obtained by the worst-case perturbed graph. Specifically, we first obtain the immune node pairs by *AdvImmune* and keep them unchanged the whole time. Then, we evaluate the transferability by node classification accuracy after defense or immunization on three settings, i.e., the clean graph, after *surrogate attack*, and after *metattack*, respectively.

We take robust training defense methods including *CEM* and *RCE* as our strong baselines. Since *metattack* uses GCN as a surrogate GNN model, we replace GCN with π-PPNP to obtain stronger attack. Besides, we adopt the Meta-Self variant of *metattack* since it is the most destructive one [19]. As for the surrogate attack, it is obtained by the perturbed graph in robustness certification. As for global budget, *metattack* can perturb 20% edges, while immune budget is only 5% edges. The settings can better demonstrate the effectiveness of our proposed *AdvImmune* method.

Table 3: The accuracy of node classification by GNN model under various attacks.

Dataset	Methods	Clean graph	Surrogate attack	metattack
Citeseer	No defense	0.74455	0.74313	0.72891
	CEM	0.71896	0.71137	0.67773
	RCE	0.73555	0.74313	0.72701
	AdvImmune	**0.74455**	**0.74550**	**0.74265**
Cora-ML	No defense	0.83701	0.79751	0.77473
	CEM	0.80925	0.77651	0.71637
	RCE	0.81957	0.77829	0.73416
	AdvImmune	**0.83701**	**0.83523**	**0.81459**
Reddit	No defense	0.90453	0.84816	0.79179
	CEM	0.86347	0.82926	0.77452
	RCE	0.86706	0.84718	0.79700
	AdvImmune	**0.90453**	**0.85598**	**0.84001**

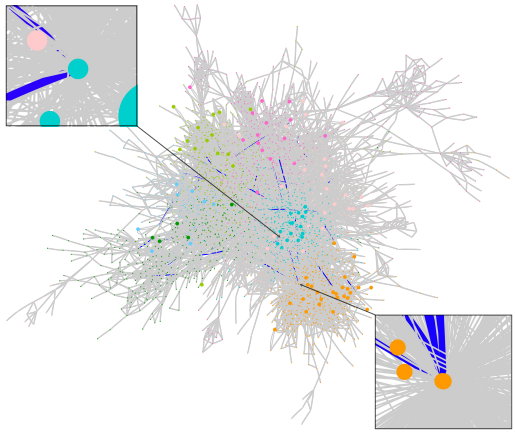
In Tab. 3, we can see that on clean graph, defense methods including *CEM* and *RCE*, even lead to a decrease of classification accuracy, which is undesirable before the adversarial attack actually happens. As for our *AdvImmune* method, it keeps the same accuracy as as original GNN model, since it only immunizes edges without changing the graph or GNN model.

Under the setting of surrogate attack, defense methods including *CEM* and *RCE* do not improve the model performance after attack, when compared with the model performance with *no defense*. This may due to the large decrease of accuracy on clean graph, making the defense less effective. As for our method, the accuracy after immunizing only 5% edges outperforms all baselines and brings significant improvement of model performance. The same immune node pairs can also significantly improve the model performance under *metattack*. These results prove that the obtained immune node pairs are transferable against various attacks, improving the model performance under attacks while maintaining the accuracy on the clean graph.

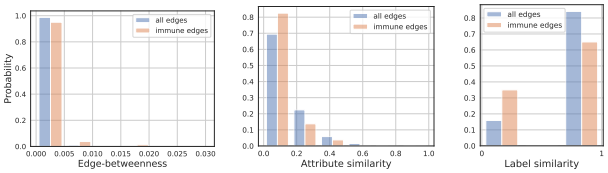
4.6 Case study

In order to have an intuitive understanding of our proposed *AdvImmune* immunization method, we take Cora-ML as a case to study the immune edges from three aspects, i.e., structure, attributes and labels. We take the scenario of **Remove-only** as an example, and local immune budget is $c_t = D_t$ and global budget is $C = 5\%$ edges. Fig. 4(a) visualizes the immune edges and the nodes that become robust through *AdvImmune* immunization. Fig. 4(b) offers the distribution of immune edges on the above three aspects.

Structure analysis. First, we take edge-betweenness as an indicator to analyze the structure characteristics of immune edges. As shown in the left figure of Fig. 4(b), the edge-betweenness of the immune edges is statistically larger than other edges. This explains why the edge-betweenness method is the strongest baseline. In



(a) Visualization of Immunization on Cora-ML



(b) Distribution of immune edges on different indicators

Figure 4: (a) The colorings of the nodes indicate different classes. Larger nodes represent nodes that become robust through immunization. The blue edges are immune edges. Insets enlarge some nodes connected to immune edges. (b) The distribution of immune edges on edge-betweenness, attribute similarity, and label similarity.

Fig. 4(a), the orange node with the highest degree, is the intersection of many immune edges. The other enlarged blue-green node connects with several immune edges, which also has high degree. This result indicates that *AdvImmune* immunization tends to immunize the edges with certain structural domination, which is critical for improving certifiable robustness of the graph.

Attribute analysis. We analyze node attribute similarity of immune edges. In the middle figure of Fig. 4(b), the attribute similarity of nodes at both ends of immune edges is mainly distributed from 0 to 0.1, while that of all edges is distributed around 0. The node attribute similarity of immune edges is lower. This may be the reason why attribute-based baselines do not work well in Fig. 3.

Label analysis. For labels, we analyze the nodes' label similarity of immune edges. There are about 80% of all edges which have the same node label at both ends, while the percentage of immune edges is only 60%. This may due to that the personalized PageRank of π-PPNP: $\mathbf{\Pi} = (1-\alpha)\left(\boldsymbol{I}_N - \alpha \boldsymbol{D}^{-1}\boldsymbol{A}\right)^{-1}$ involves the inverse operation, making the diffusion matrix be dense and the heterogeneous edges be helpful by changing the diffusion weight. It's a quite interesting finding that immunizing such heterogeneous edges are also helpful for improving the certifiable robustness of the graph, which may need to be further explored.

Experiments on Citeseer and Reddit give similar results which are not included because of space limitation.

5 RELATED WORK

GNNs have shown an exciting results on many of graph data mining tasks, e.g., node classification [14, 20, 21], network representation learning [26], graph classification [27]. However, they are proved to be sensitive to adversarial attacks [1, 28]. Attack methods can perturb both the graph structure [6] and attributes of nodes [42], while structure-based attacks are more effective, and result in more attention. Specifically, Nettack [42] attacks node attributes and the graph structure with gradient. RL-S2V [7] uses reinforcement learning to flip edges. Metattack [43] poisons the graph structure with meta-gradient. And others manipulate graph structure with approximation techniques [32] or injecting nodes [29, 33].

Various defense methods are proposed against the above attacks [18] RGCN [41] adopts Gaussian distributions as the hidden representations, which can defend against nettack [42] and RL-S2V [7]. Xu *et al.* [37] present the adversarial training framework to defend against the metattack [43]. Bayesian graph neural networks [40], trainable edge weights [34], transfer learning [30], preprocessing method [9, 19] are also used in other defense methods.

Throughout the development of graph adversarial learning, attack methods are always defended, and defense methods are also failed under the next attack. This may lead to a cat-and-mouse game, limiting the development of adversarial learning. Recently, robustness certification [22] and robust training [3, 16] methods have appeared to fill this gap. Zügner *et al.* [44] verify certifiable robustness w.r.t. perturbations on attributes. Bojchevski *et al.* [2] provide the certification w.r.t. perturbations on structures.

Different from the above research, in this paper, we firstly explore the potential and practice of adversarial immunization to improve the certifiable robustness of the graph against any admissible adversarial attack.

6 CONCLUSIONS

In this paper, we firstly propose on adversarial immunization on graphs. From the perspective of graph structure, adversarial immunization aims to improve the certifiable robustness of the graph against any admissible attack by vaccinating a fraction of node pairs, connected or unconnected. To circumvent the computationally expensive combinatorial optimization, we further propose an efficient algorithm called *AdvImmune*, which optimizes meta-gradient in a discrete way to figure out suitable immune node pairs. The effectiveness of our proposed method are evaluated on three well-known network datasets, including two citation networks and one social network. Experimental results reveal that our *AdvImmune* approach significantly improves the certifiable robustness of the graph. We also verify the generality and transferability of *AdvImmune* under various attacks. However, since our method is based on the robustness certification, it can only be applied for PPNP-like GNN models. Also, the complexity of PPNP limits its application to large graphs. We tend to solve these challenges and explore the adversarial immunization focusing on both graph structures and node attributes in the future work.

REFERENCES

[1] Aleksandar Bojchevski and Stephan Günnemann. 2019. Adversarial Attacks on Node Embeddings via Graph Poisoning. In *Proceedings of the 36th International Conference on Machine Learning (ICML '19)*. 695–704.

[2] Aleksandar Bojchevski and Stephan Günnemann. 2019. Certifiable Robustness to Graph Perturbations. In *Advances in Neural Information Processing Systems 32*. 8319–8330.

[3] Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. 2020. Efficient Robustness Certificates for Discrete Data: Sparsity-Aware Randomized Smoothing for Graphs, Images and More. In *Proceedings of the 37th International Conference on Machine Learning (ICML '20)*. 11647–11657.

[4] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. 2020. Popularity Prediction on Social Platforms with Coupled Graph Neural Networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 70–78.

[5] Nicholas Carlini and David A. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP '17)*.

[6] Liang Chen, Jintang Li, Jiaying Peng, Tao Xie, Zengxu Cao, Kun Xu, Xiangnan He, and Zibin Zheng. 2020. A Survey of Adversarial Learning on Graphs. *ArXiv* abs/2003.05730 (2020).

[7] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial Attack on Graph Structured Data. In *Proceedings of the 35th International Conference on Machine Learning (ICML '18)*. 1123–1132.

[8] Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang. 2019. Adversarial Training Methods for Network Embedding. In *Proceedings of The Web Conference 2019 (WWW '19)*. 329–339.

[9] Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. 2020. All You Need Is Low (Rank): Defending Against Adversarial Attacks on Graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 169–177.

[10] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference (WWW '19)*. 417–426.

[11] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML '17)*. 1126–1135.

[13] Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99, 12 (2002), 7821–7826.

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*. 1024–1034.

[15] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. 2018. Adaptive Sampling Towards Fast Graph Representation Learning. In *Advances in Neural Information Processing Systems 31*. 4558–4567.

[16] Jinyuan Jia, Binghui Wang, Xiaoyu Cao, and Neil Zhenqiang Gong. 2020. Certified Robustness of Community Detection against Adversarial Structural Perturbation via Randomized Smoothing. In *Proceedings of The Web Conference 2020 (WWW '20)*. 2718–2724.

[17] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. [n.d.]. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment.. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[18] Wei Jin, Yaxin Li, Han Xu, Yiqi Wang, and Jiliang Tang. 2020. Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study. *ArXiv* abs/2003.00653 (2020).

[19] Wei Jin, Yao Ma, Xiaorui Liu, Xian-Feng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*.

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR '17)*.

[21] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations (ICLR '19)*.

[22] Yang Liu, Xianzhuo Xia, Liang Chen, X. He, Carl Yang, and Z. Zheng. 2020. Certifiable Robustness to Discrete Adversarial Perturbations for Factorization Machines. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 419–428.

[23] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report.

[24] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. DeepInf: Social Influence Prediction with Deep Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 2110–2119.

[25] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. In *International Conference on Learning Representations (ICLR '18)*.

[26] Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. 2017. Struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '17)*. 385–394.

[27] Bastian Rieck, Christian Bock, and Karsten Borgwardt. 2019. A persistent weisfeiler-lehman procedure for graph classification. In *Proceedings of the 36th International Conference on Machine Learning (ICML '19)*. 5448–5458.

[28] Lichao Sun, Ji Wang, Philip S. Yu, and Bo Li. 2018. Adversarial Attack and Defense on Graph Data: A Survey. *ArXiv* abs/1812.10528 (2018).

[29] Yiwei Sun, Suhang Wang, Xian-Feng Tang, Tsung-Yu Hsieh, and Vasant G Honavar. 2020. Adversarial Attacks on Graph Neural Networks via Node Injections: A Hierarchical Reinforcement Learning Approach. In *Proceedings of The Web Conference 2020 (WWW '20)*. 673–683.

[30] Xian-Feng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2020. Transferring Robustness for Graph Neural Network Against Poisoning Attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 600–608.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR '18)*.

[32] Binghui Wang and Neil Zhenqiang Gong. 2019. Attacking Graph-Based Classification via Manipulating the Graph Structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. 2023–2040.

[33] Jihong Wang, Minnan Luo, Fnu Suya, Jundong Li, Zijiang Yang, and Qinghua Zheng. 2020. Scalable Attack on Graph Data by Injecting Vicious Nodes. *arXiv preprint arXiv:2004.13825* (2020).

[34] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial Examples for Graph Data: Deep Insights into Attack and Defense. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. 4816–4823.

[35] Bingbing Xu, Huawei Shen, Qi Cao, Keting Cen, and Xueqi Cheng. 2019. Graph convolutional networks using heat kernel for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. 1928–1934.

[36] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. 2019. Graph Wavelet Neural Network. In *International Conference on Learning Representations (ICLR '19)*.

[37] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. 3961–3967.

[38] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2020. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *International Conference on Learning Representations (ICLR '20)*.

[39] Yingxue Zhang, S Khan, and Mark Coates. [n.d.]. Comparing and detecting adversarial attacks for graph deep learning. In *Proc. Representation Learning on Graphs and Manifolds Workshop, Int. Conf. Learning Representations, New Orleans, LA, USA (RLGM @ ICLR '19)*.

[40] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Üstebay. [n.d.]. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

[41] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2019. Robust Graph Convolutional Networks Against Adversarial Attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 1399–1407.

[42] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 2847–2856.

[43] Daniel Zügner and Stephan Günnemann. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations (ICLR '19)*.

[44] Daniel Zügner and Stephan Günnemann. 2019. Certifiable Robustness and Robust Training for Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 246–256.

[45] Daniel Zügner and Stephan Günnemann. 2020. Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*.