Flexible Noise Based Robustness Certification Against Backdoor Attacks in Graph Neural Networks

Hiroya Kato[1,*], Ryo Meguro[2,*], Seira Hidano[1], Takuo Suganuma[2] and Masahiro Hiji[2]
[1]*KDDI Research, Inc., Saitama, Japan*
[2]*Tohoku University, Miyagi, Japan*

Keywords: Graph Neural Networks, Robustness Certification, Backdoor Attacks, AI Security.

Abstract: Graph neural networks (GNNs) are vulnerable to backdoor attacks. Although empirical defense methods against such attacks are effective to some extent, they may be bypassed by adaptive attacks. Thus, recently, robustness certification that can certify the model robustness against any type of attack has been proposed. However, existing certified defenses have two shortcomings. The first one is that they add uniform defensive noise to the entire dataset, which degrades the robustness certification. The second one is that unnecessary computational costs for data with different sizes are required. To address them, in this paper, we propose flexible noise based robustness certification against backdoor attacks in GNNs. Our method can flexibly add defensive noise to binary elements in an adjacency matrix with two different probabilities. This leads to improvements in the model robustness because the defender can choose appropriate defensive noise depending on datasets. Additionally, our method is applicable to graph data with different sizes of adjacency matrices because a calculation in our certification depends only on the size of attack noise. Consequently, computational costs for the certification are reduced compared with a baseline method. Our experimental results on four datasets show that our method can improve the level of robustness compared with a baseline method. Furthermore, we demonstrate that our method can maintain a higher level of robustness with larger sizes of attack noise and poisoning.

1 INTRODUCTION

Graph neural networks (GNNs) have drawn attention for their ability to classify graph-structured data such as social networks and molecular structures. However, as with the cases where general machine learning models are vulnerable to attacks such as evasion, poisoning, and backdoor attacks (Goodfellow et al., 2014; Shafahi et al., 2018; Gu et al., 2019), GNNs also exhibit the same vulnerabilities (Zügner et al., 2018; Kwon et al., 2019; Chen et al., 2020; Jiang et al., 2022; Zhang et al., 2021; Meguro et al., 2024). To be specific, even a slight alternation of edge information can change node or graph labels. Various empirical defense methods are proposed to counter these attacks (Wang et al., 2019; Zhang and Zitnik, 2020; Zhang et al., 2020; Jiang and Li, 2022).

Many of these methods aim to improve the robustness of models or detect poisoned data by heuristically analyzing vulnerable parts or the characteristics of individual poisoned data. However, these methods can be circumvented by clever attackers. If data which satisfy certain conditions are regarded as poison by the defender, intentionally crafted poison which avoids those conditions cannot be detected. Consequently, such defenses lack guarantees of robustness, which means they remain vulnerable to unknown or adaptive attacks. To address any type of attacks, some researchers develop methods that add consistent Gaussian or probabilistic defensive noise to the entire data (Jia et al., 2019; Cohen et al., 2019; Wang et al., 2021; Weber et al., 2023; Zhang et al., 2022). These methods, known as robustness certification via randomized smoothing, can theoretically certify robustness against any type of attack. They calculate a certified radius, which defines a specific range within which the model consistently produces the same prediction for a perturbed sample. Robustness certification is widely studied, particularly in the field of image classification, against evasion, poisoning, and backdoor attacks. On the other hand, robustness guarantees for classifiers that work with graph data are not fully established, especially in poisoning

*Equal contributions.

Kato, H., Meguro, R., Hidano, S., Suganuma, T. and Hiji, M.
Flexible Noise Based Robustness Certification Against Backdoor Attacks in Graph Neural Networks.
DOI: 10.5220/0013188700003899

In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 2*, pages 552-563
ISBN: 978-989-758-735-1; ISSN: 2184-4356

and backdoor attacks. Graph classifiers are expected to be used in security-critical areas such as recommendation systems and malware detection (Liu et al., 2020; Qiu et al., 2020; Yang et al., 2021; Guo et al., 2021; Feng et al., 2020).

Therefore, providing robustness certification is crucial to ensuring the safety of GNNs. Considering the fact that edge information on graphs is binary data, existing methods (Wang et al., 2021; Zhang et al., 2022) are promising to achieve robustness certification against graph backdoor attacks. Thus, our method is mainly inspired by the concepts and techniques introduced in them.

However, we argue that there are two challenges when they are directly utilized for robustness certification against graph backdoor attacks. The first is that these methods apply defensive noise to all elements in the adjacency matrices with a certain probability. In such a situation, the level of robustness in the graph data can be significantly reduced. This is because the difference in importance between existing edges and non-existent ones is not considered at all whereas they have different importance. The second is that the existing method (Zhang et al., 2022) refers to the data size when calculating the certified radius. This limitation is not favorable for graph datasets because it is difficult to align the size of adjacency matrices, which incurs unnecessary computational costs.

To address them, we propose flexible noise based robustness certification for GNNs (or binary data classifiers in general) against backdoor attacks. Our method can achieve a higher level of robustness by realizing more flexible certification against backdoor attacks. Our method individually sets the different noise probability for elements of 0 and 1 in adjacency matrices. This leads to improvements in model performance and robustness because the defender can choose the appropriate defensive noise from a large set of parameters. Additionally, our method is designed to depend only on the size of attack noise when calculating the certified radius. This is why our method is applicable to graph data with any number of nodes. Our contributions are summarized as follows:

- We propose a robustness certification method against backdoor attacks that can flexibly add defensive noise to binary data with two different probabilities.
- We demonstrate the calculation of the certified radius when flexible noise is applied to elements of binary data to achieve flexible robustness certification.
- Our method is applicable to datasets composed of graph data with different sizes and reduces computational costs for the certification.
- Our experimental results show that adding defensive noise to binary data with two different probabilities is effective in improving the level of robustness. Additionally, our method can maintain a certain level of robustness with larger sizes of attack noise and poisoning sizes compared with a baseline method.

2 RELATED WORK

Robustness Certification via Gaussian Noise: Cohen et al. (Cohen et al., 2019) propose a certified defense against evasion attacks using a technique called randomized smoothing for image classifiers. This method ensures that the model consistently produces correct predictions on adversarial examples regardless of the type of attack if attack noise is below a certain threshold. It is a groundbreaking approach that ends the cat-and-mouse game between attackers and defenders. In that certified defense, defensive noise based on the normal distribution is added to an image to obtain multiple noisy images. These noisy images are input into an image classifier and outputs are utilized to calculate the certified radius. The basic idea behind that method is that the prediction of the smoothed model remains consistent, even if an attacker adds a small amount of attack noise to benign data, thus achieving robustness certification.

Weber et al. (Weber et al., 2023) propose a method that extends the guarantees provided by (Cohen et al., 2019), not only to evasion attacks but also to backdoor attacks in the image domain. In that method, defensive noise is added to a training dataset to offer robustness guarantees against backdoor attacks. That method ensures that the predictions of a backdoored model are the same as a benign model. Note that this consistency is guaranteed only when the total size of the triggers injected into the training dataset remains below a certain threshold. Additionally, to maintain model performance, they add noise to test data based on the hash value of the model. This helps bring the distribution of the test data closer to the distribution of the training dataset, leading to high model performance. However, robustness certification via Gaussian noise is not applicable to binary data such as graph data.

Robustness Certification via Probabilistic Noise: Jia et al. (Jia et al., 2021) theoretically demonstrate certified robustness of a model trained using an ensemble learning method called Bagging. In Bagging, the operation of randomly selecting a portion of the training dataset is repeated N times to create N subsample datasets. Then, a single model is trained with

each subsample dataset, resulting in N trained models. The label for a test sample is determined using the prediction results of the N models. In a model trained with Bagging, even if a small amount of poisoned data are included in the overall dataset, the model is robust against poisoning attacks. This is because the effect of the poison can be mitigated by using relatively small, randomly selected training subsamples. That method does not strictly add noise to the data. However, we include that method in this category because the operation of randomly selecting data of subsamples is a probabilistic masking applied to the entire dataset.

Wang et al. (Wang et al., 2021) propose a certified defense against evasion attacks for binary data classifiers, including GNNs. Unlike the method described above, that approach utilizes probabilistic defensive noise. That method mitigates the effect of adversarial examples by altering each element of 0 or 1 with a certain probability. That method utilizes only straightforward probability calculations and Neyman-Pearson lemma, making it an approach that could inspire extensions to discrete data classifiers in general.

Zhang et al. (Zhang et al., 2022) propose a robustness certification that addresses a wide range of attack types against any discrete data classifier. That method uses a defensive noise that changes each element of the data, such as pixels in an image, to a different value with a certain probability. The defensive noise is applied to both the training dataset and the test data, extending the guarantee coverage not only to evasion attacks but also to poisoning and backdoor attacks. That method incorporates the approach proposed by (Jia et al., 2021) and integrates ensemble learning. Therefore, that method offers more robust guarantees against poisoning and backdoor attacks. Additionally, the use of recurrence relations reduces computational complexity and provides guarantees in practical time frames, contributing to the practicality.

Considering the fact that edge information in graphs is binary data, the previous methods (Wang et al., 2021; Zhang et al., 2022) are promising for realizing robustness certification against graph backdoor attacks. Thus, our method is mainly inspired by the concepts and techniques introduced in the previous methods. Our method also utilizes the Neyman-Pearson lemma and Monte Carlo sampling to derive a more practical condition of the robustness certification.

Limitations of Existing Certified Defenses: The previous methods are competent robustness certification via randomized smoothing. However, there are two challenges associated with these methods. First, these methods apply defensive noise to all elements of data with a certain probability. Although applying defensive noise at a single level can guarantee the security of any data pixel, the characteristics of the data are also lost with a certain probability regardless of their importance. This may lead to a decrease in the performance of models. The second challenge, particularly in robustness certification for discrete data classifiers, is that the method (Zhang et al., 2022) refers to the data size when calculating the certified radius. This is not a problem for image datasets because they can easily be resized to a fixed size. However, for datasets such as graph datasets, which are difficult to resize, it is necessary to apply different calculations to each differently sized data, entailing unnecessary computational costs.

On the other hand, our method addresses the inflexibility of the defensive noise and the problems related to data size.

3 PROBLEM DEFINITION

3.1 Threat Model

In this work, we assume that a GNN model is utilized for graph classification task, which assigns a label to the entire graph data. Additionally, we assume that an attacker has access to the entire training dataset D_{entire} which consists of adjacency matrices. Therefore, we limit the variables of the GNN model to the adjacency matrices and assume that all other elements, such as node features and graph labels, remain constant. Therefore, an objective of the backdoor attack is formulated as

$$\text{if } f(x, D_{\text{entire}}) = l_A, \text{then } f(x \oplus \delta, D_{\text{entire}} \oplus \Delta) = l_B, \tag{1}$$

where $l_A \neq l_B$ and f is a graph data classifier. δ is attack noises inserted into a test data x, and Δ is a set of attack noises $\{\delta_i | 1 \leq i \leq p\}$ which are inserted into p training data. $\oplus$ represents the exclusive OR (XOR). This backdoor attack is an attack where the attacker modifies a portion of the training dataset and the data they want to misclassify during inference by embedding a common marker called a trigger. By training the model with data containing the trigger, the model becomes more likely to respond to the trigger, allowing the attacker to manipulate the inference results.

The attacker sets an attack budget to avoid detection. In this case, we assume that the attacker sets two attack budgets B_{poison} and B_{noise}. The first is the amount of poison p, mixed into the training dataset. The second is the size of the attack noise δ, added to each poisoned data (either in the poisoned training data or the poison used during the test phase). Gener-

ally, the smaller the values of p and δ, the more difficult it becomes to detect the attack. Then, the objective of avoiding detection is described as

$$\begin{aligned} & p \leq B_{\text{poison}} \text{ and } \|\delta\|_0 \leq B_{\text{noise}} \\ & \text{where } p = \sum_{i=1}^{|D_{\text{entire}}|} \mathbb{I}[d_{\text{entire},i} \neq \tilde{d}_{\text{entire},i}], \\ & d_{\text{entire},i} \in D_{\text{entire}}, \tilde{d}_{\text{entire},i} \in \tilde{D}_{\text{entire}}. \end{aligned} \tag{2}$$

Note that $\tilde{D}_{\text{entire}} = D_{\text{entire}} \oplus \Delta$, which is a poisoned dataset.

3.2 Defense Goal

The goal of the defender in robustness certification against backdoor attacks is to construct a robust model g, that ensures if the classifier's prediction for data without attack noise is l_A, then the prediction label for the data with attack noise below a certain threshold is also l_A. Let x and y denote a benign and poisoned test sample, respectively. The above goal is formulated as

$$\begin{aligned} & \text{if } P(g(x,D) = l_A) > P(g(x,D) = l_B), \\ & \text{then } P(g(y,\tilde{D}) = l_A) > P(g(y,\tilde{D}) = l_B) \\ & \text{where } y = x \oplus \delta \ \wedge \ D \subseteq D_{\text{entire}} \ \wedge \ \tilde{D} \subseteq \tilde{D}_{\text{entire}} \end{aligned} \tag{3}$$

for any label $l_B \neq l_A$. Furthermore, for sets A and B, $A \oplus B$ represents the set obtained by taking the XOR of each element in the sets. We use an ensemble learning method to construct a robust model. Therefore, subsample $D \subseteq D_{\text{entire}}$ and $\tilde{D} \subseteq \tilde{D}_{\text{entire}}$ are utilized to train g. Let D and $\tilde{D}$ denote $\{d_i | 1 \leq i \leq |D|\}$ and $\{\tilde{d}_i | 1 \leq i \leq |D|\}$, respectively. Additionally, $y = x \oplus \delta$, $\tilde{D} = D \oplus \Delta$ where $\Delta = \{\delta_i | 1 \leq i \leq |D|\}$ and $g(x,D) = f(x \oplus \varepsilon, D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\})$, where ε and ε_i are realization of $\Upsilon = \{A_i \sim \text{Bernoulli}(1-\beta_{x_i}) \mid 1 \leq i \leq |x|\}$ and $\Upsilon_i = \{A_j \sim \text{Bernoulli}(1-\beta_{d_{i,j}}) \mid d_i \in D \wedge 1 \leq j \leq |d_i|\}$.

Under this assumption, the defender derives the maximum attack noise size or the maximum poisoning size which satisfy Eq.(3). Note that the maximum attack noise size refers to the maximum norm of δ added to a data. The maximum poisoning size is the maximum number of poisoned data that are inserted into D_{entire}.

Practically, it is difficult to calculate the probabilities shown in Eq.(3) that the classifier outputs a specific label. Therefore, we use Monte Carlo sampling to calculate the lower and upper bounds of $P(g(x,D) = l_A)$ and $P(g(x,D) = l_B)$. Then, we use Neyman-Pearson lemma for binary random variables shown in (Wang et al., 2021) to calculate the lower and upper bounds of $P(g(y,D) = l_A)$ and $P(g(y,D) = l_B)$. The Neyman-Pearson lemma is as follows.

- $\exists r > 0, S_1 = \left\{ z \in \{0,1\}^n \mid \frac{P(X=z)}{P(Y=z)} > r \right\}, S_2 = \left\{ z \in \{0,1\}^n \mid \frac{P(X=z)}{P(Y=z)} = r \right\}$.
  Assume $S_3 \subseteq S_2 \wedge S_{\text{benign}} = S_1 \cup S_3$.
  If $P(h(X) = 1) \geq P(X \in S_{\text{benign}})$, then $P(h(Y) = 1) \geq P(Y \in S_{\text{benign}})$.
- $\exists r > 0, S_1 = \left\{ z \in \{0,1\}^n \mid \frac{P(X=z)}{P(Y=z)} < r \right\}, S_2 = \left\{ z \in \{0,1\}^n \mid \frac{P(X=z)}{P(Y=z)} = r \right\}$.
  Assume $S_3 \subseteq S_2 \wedge S_{\text{poison}} = S_1 \cup S_3$.
  If $P(h(X) = 1) \leq P(X \in S_{\text{poison}})$, then $P(h(Y) = 1) \leq P(Y \in S_{\text{poison}})$.

X and Y are random variables obtained after defensive noise Υ is added to x and y, respectively. The Neyman-Pearson lemma expresses the prediction probability of the classifier as the probability that X and Y are included in a specific set S_{benign} and S_{poison}, making the calculation relatively straightforward. Now, we apply the Neyman-Pearson lemma to the problem of ensuring the robustness of a classifier against backdoor attacks. We define $h(X) = \mathbb{I}[g(X) = l_A]$. Let $\mathbb{I}[Q]$ be 1 if the proposition Q is true, and 0 otherwise. Through this lemma, we can derive a new condition from Eq.(3) as

$$\begin{aligned} & \text{if } P(X \in S_{\text{benign}}) > P(X \in S_{\text{poison}}) \text{ holds}, \\ & \text{then } P(Y \in S_{\text{benign}}) > P(Y \in S_{\text{poison}}) \text{ also holds} \end{aligned} \tag{4}$$

if we define the S_{benign} and S_{poison}, appropriately.

4 PROPOSAL

4.1 Overview

We propose flexible noise based robustness certification against backdoor attacks in GNNs. Our approach is summarized into two main ideas. First, our method individually sets the different noise probability for elements of 0 and 1 in adjacency matrices. This leads to improvements in model performance and robustness because the defender can choose appropriate defensive noise from a large set of parameters. Second, our method is applicable to graph data with any number of nodes (i.e., large adjacency matrices). This is because the calculation of the likelihood ratio in our certification depends only on the size of attack noise.

In what follows, we describe the algorithm of our certification in the training phase and the testing one in detail. Afterward, theoretical calculation of our method is explained in detail.

4.2 Algorithm of Proposed Robustness Certification

In this section, we explain the algorithm on how to build a robust model. The outline of this algorithm is shown in Figure 1.

Algorithm in the Training Phase. First, we describe the training phase of a binary data classifier. This section corresponds to lines 1–4 of the Algorithm 1. Our method randomly selects e data from the entire training dataset D_{entire} to create a small subsample dataset D. This subsample extraction is performed independently N times, resulting in N different subsample datasets. Furthermore, defensive noises are added to the features of the data in the subsamples. Here, among the features of the data, elements of 0 are retained with a probability of β_0 (i.e., they are changed to 1 with probability $1-\beta_0$), and elements of 1 are retained with a probability of β_1 (i.e., they are changed to 0 with probability $1-\beta_1$). Each of these randomized subsamples is described as $D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\}$. After that, our method prepares N classifiers. Each of these N models is trained on its respective randomized subsample and is referred to as $f(D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\})$.

Algorithm in the Testing Phase. This section corresponds to lines 5–12 in the Algorithm 1. First, our method adds defensive noise to the features of each data in the test dataset, similar to the training phase. Then, the test data x is predicted N times using the N models. As a result, we obtain N outputs $l = f(x \oplus \varepsilon, D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\})$. It is possible to statistically estimate the top 1 and top 2 label probabilities by counting N_{l_A} and N_{l_B}, which are the number of top 1 and top 2 labels in the N outputs, respectively. Finally, based on these results, we derive the certified radius to theoretically determine the range in which the model's predictions remain consistent even when attack noise is added to the data. The specific theoretical calculation to obtain the certified radius is discussed in the following sections.

4.3 Theoretical Calculation

In this section, we describe how to compute the certified radius using our method to defend against backdoor attacks. To define S_{benign} and S_{poison} in Eq.(4), we first calculate $P(X = z \wedge D_\varepsilon = D_z)$ and $P(Y = z \wedge \tilde{D}_\varepsilon = D_z)$. Then, we obtain the likelihood ratio $\frac{P(X=z \wedge D_\varepsilon = D_z)}{P(Y=z \wedge \tilde{D}_\varepsilon = D_z)}$. Note that $z \in \{0,1\}^{|x|}$ and $D_z = \{d_{z,i} \in \{0,1\}^{|d_{z,i}|} | 1 \leq i \leq |D|\}$ are elements that could potentially appear after the addition of defensive noise. We focus on the points where the attack noise is introduced and proceed to calculate $\frac{P(X=z \wedge D_\varepsilon = D_z)}{P(Y=z \wedge \tilde{D}_\varepsilon = D_z)}$. Our method retains elements of 0 in features of test and training data with probability β_0, and elements of 1 with probability β_1. Under this premise, we compute $P(X = z \wedge D_\varepsilon = D_z)$ and $P(Y = z \wedge \tilde{D}_\varepsilon = D_z)$ where D_ε and $\tilde{D}_\varepsilon$ are $D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\}$ and $\tilde{D} \oplus \{\varepsilon_i | 1 \leq i \leq |D|\}$, respectively. Among the elements of 0 in $x_{\text{tmp}} \in \{\{x\} \cup D\}$, considering the positions where the attack noise is introduced, $m_{0,i}$ elements of x_{tmp} and $z_{\text{tmp}} \in \{\{z\} \cup D_z\}$ are the same, while $\|\delta_0\|_0 - m_{0,i}$ elements are different. $m_{0,0}$ and $m_{0,i} (1 \leq i \leq |D|)$ are assigned to x and $d_i (1 \leq i \leq |D|)$, respectively. Similarly, among the elements of 1 in x_{tmp}, $m_{1,i}$ elements of x_{tmp} and z_{tmp} are the same, while $\|\delta_1\|_0 - m_{1,i}$ elements are different. These $m_{i,j} (i \in \{0,1\}, 1 \leq j \leq |D|)$ take values in range of $[0, \|\delta_i\|_0]$. δ_0 and δ_1 are the attack noises that flip the elements of 0 and 1 in x_{tmp}, respectively. Then, $P(X = z \wedge D_\varepsilon = D_z)$ and $P(Y = z \wedge \tilde{D}_\varepsilon = D_z)$ are calculated as

$$P(X = z \wedge D_\varepsilon = D_z) = \prod_{i=0}^{c} \beta_0^{m_{0,i}} (1-\beta_0)^{\|\delta_0\|_0 - m_{0,i}} \beta_1^{m_{1,i}} (1-\beta_1)^{\|\delta_1\|_0 - m_{1,i}} \cdot p_i \tag{5}$$

$$P(Y = z \wedge \tilde{D}_\varepsilon = D_z) = \prod_{i=0}^{c} (1-\beta_1)^{m_{0,i}} \beta_1^{\|\delta_0\|_0 - m_{0,i}} (1-\beta_0)^{m_{1,i}} \beta_0^{\|\delta_1\|_0 - m_{1,i}} \cdot p_i. \tag{6}$$

p_i is the probability that, in areas unaffected by attack noise, the randomized result of x_{tmp} matches z_{tmp}. The

Algorithm 1: Computing a certified radius.

Input : Train dataset D_{entire}, test data x
Output: Certified radius R_x for x

```
// Training
for i = 1 to N do
    D ← sample e data from D_entire
    D ← D ⊕ {ε_i | 1 ≤ i ≤ |D|}
    f_i ← train(f, D)
// Testing
counter ← (0)_{i=1}^{L}
for i = 1 to N do
    l ← f_i(x ⊕ ε)                ▷ Label Prediction
    counter[l] ← counter[l] + 1
N_{l_A} ← counter[l_A]
N_{l_B} ← counter[l_B]
calculate p_A and p_B on the basis of Eq.(19)
R_x ← maximum p or ||δ||_0 satisfying Eq.(4)
```

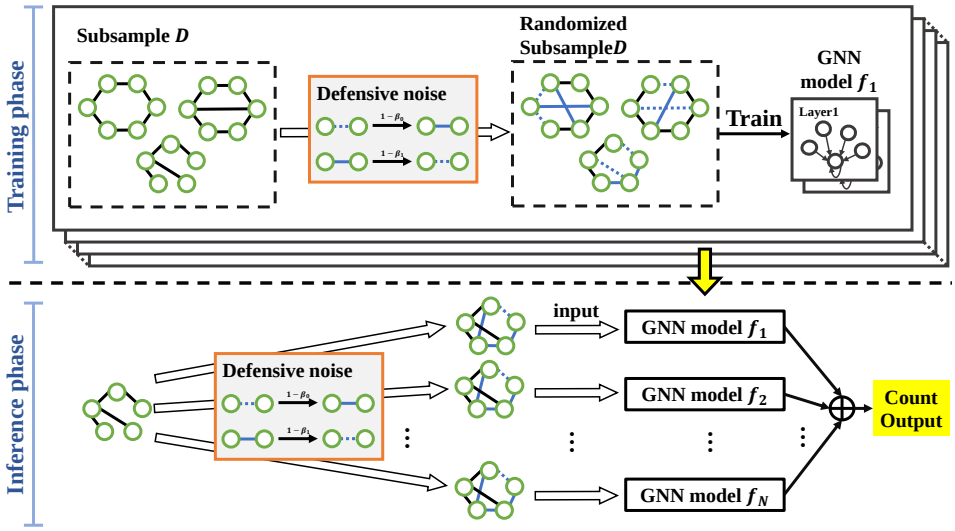


Figure 1: Overview of constructing a robust model through our method.

likelihood ratio is formulated as

$$\frac{P(X = z \wedge D_\varepsilon = D_z)}{P(Y = z \wedge \tilde{D}_\varepsilon = D_z)} = C^{c+1} \cdot \left(\frac{\beta_0}{1-\beta_0}\frac{\beta_1}{1-\beta_1}\right)^{\sum_{i=0}^{c}(m_{0,i}+m_{1,i})} \quad (7)$$

$$\text{where } C = \frac{(1-\beta_0)^{\|\delta_0\|_0}(1-\beta_1)^{\|\delta_1\|_0}}{\beta_0^{\|\delta_1\|_0}\beta_1^{\|\delta_0\|_0}}.$$

There are $2c+2$ variables, $m_{0,i}$ and $m_{1,i}$. However, since they are consolidated in the exponent, we treat their sum as a single variable. That is, we introduce a new variable m, where $m = \sum_{i=0}^{c}(m_{0,i}+m_{1,i})$. This m represents the total number of different elements between (x,D) and (z,D_z). Based on this likelihood ratio, we define a fundamental set $R(c,m)$ as

$$R(c,m) = \left\{ (z,D_z) \;\middle|\; \sum_{i=1}^{|D|} \mathbb{I}[D_{\varepsilon,i} \neq \tilde{D}_{\varepsilon,i}] = c \wedge \frac{P(X = z \wedge D_\varepsilon = D_z)}{P(Y = z \wedge \tilde{D}_\varepsilon = D_z)} = r(c,m) \right\},$$

$$\text{where } r(c,m) = C^{c+1} \cdot \left(\frac{\beta_0}{1-\beta_0}\frac{\beta_1}{1-\beta_1}\right)^{m}. \quad (8)$$

We construct the set S_{benign} based on this basic set. Considering the premise of $P(h(X) = 1) \geq P(X \in S_{\text{benign}})$, the set S_{benign} represents the lower bound of the probability that the classifier h returns the correct probability for benign data. Therefore, $P(X \in S_{\text{benign}}) = \underline{p_A}$, where $\underline{p_A}$ is the lower bound of the probability that the classifier outputs label l_A. Calculating $P(h(X) = 1)$ directly requires complex computations depending on the model h. However, the lower bound can actually be estimated relatively easily using Monte Carlo sampling. While the specific method for calculating this lower bound is described later, we proceed with the assumption that $\underline{p_A}$ is a known value.

We build S_{benign} on the basis of $R(c,m)$ from here. Let R_i be sorted in descending order according to the likelihood ratio $r(c,m)$, and S_{benign} is defined as

$$S_{\text{benign}} = \bigcup_{1 \leq i \leq a^*} R_i \cup R_{\text{sub}},$$
$$s.t.\ a^* = \operatorname*{argmax}_a \sum_{i=1}^{a} P((X,D_\varepsilon) \in R_i) \leq \underline{p_A}. \quad (9)$$

S_{sub} is a set defined as

$$S_{\text{sub}} \subseteq R_{a^*+1} \wedge P((X,D_\varepsilon) \in S_{\text{sub}}) = \underline{p_A} - \sum_{i=1}^{a^*} P((X,D_\varepsilon) \in R_i). \quad (10)$$

In the Neyman-Pearson lemma, S_{sub} corresponds to the set S_3 and $P((X,D_\varepsilon) \in S_{\text{benign}})$ is adjusted so that it exactly matches $\underline{p_A}$. S_{benign} is a set that selects a portion of data with highest likelihood ratio. S_{benign} can be interpreted as a collection of samples that appear to be the most benign.

We can now compute $P((X,D_\varepsilon) \in S_{\text{benign}})$ and $P((Y,\tilde{D_\varepsilon}) \in S_{\text{benign}})$. Obviously, $P((X,D_\varepsilon) \in S_{\text{benign}})$ is formulated as

$$\sum_{i=1}^{a^*} P((X,D_\varepsilon) \in R_i) + P((X,D_\varepsilon) \in R_{\text{sub}}) = \underline{p_A}. \quad (11)$$

Then, we compute $P((Y,\tilde{D_\varepsilon}) \in S_{\text{benign}})$. $P((Y,\tilde{D_\varepsilon}) \in S_{\text{benign}})$ represents the probability that, when defensive noise is added to poisoned data, it matches the

most benign-looking data, (z, D_z), in S_{benign}. Similar to the case of $P((X, D_\varepsilon) \in S_{\text{benign}})$, $P((Y, \tilde{D_\varepsilon}) \in S_{\text{benign}})$ is formulated as

$$\sum_{i=1}^{a^*} P((Y, \tilde{D_\varepsilon}) \in R_i) + \frac{P((X, D_\varepsilon) \in R_{\text{sub}})}{r_{a^*+1}}. \quad (12)$$

The final term is divided by the likelihood ratio r_{a^*+1} of S_{sub}. This is easier to understand when recalling the definition of the likelihood ratio, and can be derived from $r_{a^*+1} = \frac{P(X=z \wedge D_\varepsilon = D_z)}{P(Y=z \wedge \tilde{D_\varepsilon} = D_z)}$ to

$$\begin{aligned}
&\frac{P((X, D_\varepsilon) \in R_{\text{sub}})}{r_{a^*+1}} \\
&= \frac{P((X, D_\varepsilon) \in R_{\text{sub}})}{\frac{P(X=z \wedge D_\varepsilon = D_z)}{P(Y=z \wedge \tilde{D_\varepsilon} = D_z)}} \\
&= P(Y = z \wedge \tilde{D_\varepsilon} = D_z) \cdot \frac{P((X, D_\varepsilon) \in R_{\text{sub}})}{P((X, D_\varepsilon) \in R_{\text{sub}})} \\
&= P(Y = z \wedge \tilde{D_\varepsilon} = D_z) \cdot \frac{|R_{\text{sub}}| \cdot P((X, D_\varepsilon) \in R_{\text{sub}})}{P((X, D_\varepsilon) \in R_{\text{sub}})} \\
&= P(Y = z \wedge \tilde{D_\varepsilon} = D_z) \cdot |R_{\text{sub}}| \\
&= P((Y, \tilde{D_\varepsilon}) \in R_{\text{sub}}).
\end{aligned} \quad (13)$$

This term, unlike the others involving $P((Y, \tilde{D_\varepsilon}) \in R_i)$, forms a truncated set, which makes it mathematically difficult to count. Therefore, it is simply expressed in a form using the more easily obtainable $P((X, D_\varepsilon) \in R_{\text{sub}})$.

Next, we calculate $P((X, D_\varepsilon) \in S_{\text{benign}})$ and $P((Y, \tilde{D_\varepsilon}) \in S_{\text{benign}})$ using combinatorial formulas and exponentiation. Given that R_i is a set of rearranged $R(c, m)$, it suffices to compute $P((X, D_\varepsilon) \in R(c, m))$ and $P((Y, \tilde{D_\varepsilon}) \in R(c, m))$ for any non-negative integer m. Therefore, we proceed with these calculations.

Then, $P((X, D_\varepsilon) \in R(c, m))$ and $P((Y, \tilde{D_\varepsilon}) \in R(c, m))$ are formulated as

$$\begin{aligned}
&P((X, D_\varepsilon) \in R(c, m)) = \\
&\frac{\binom{p}{c}\binom{n-p}{e-c}}{\binom{n}{e}} \sum_{\sum_{i=0}^{c}(m_{0,i}+m_{1,i})=m} \prod_{i=0}^{c} \left\{ \binom{\|\delta_0\|_0}{m_{0,i}} \binom{\|\delta_1\|_0}{m_{1,i}} \times \right. \\
&\left. \beta_0^{m_{0,i}} (1-\beta_0)^{\|\delta_0\|_0 - m_{0,i}} \beta_1^{m_{1,i}} (1-\beta_1)^{\|\delta_1\|_0 - m_{1,i}} \right\},
\end{aligned} \quad (14)$$

$$\begin{aligned}
&P((Y, \tilde{D_\varepsilon}) \in R(c, m)) = \\
&\frac{\binom{p}{c}\binom{n-p}{e-c}}{\binom{n}{e}} \sum_{\sum_{i=0}^{c}(m_{0,i}+m_{1,i})=m} \prod_{i=0}^{c} \left\{ \binom{\|\delta_0\|_0}{m_{0,i}} \binom{\|\delta_1\|_0}{m_{1,i}} \times \right. \\
&\left. (1-\beta_1)^{m_{0,i}} \beta_1^{\|\delta_0\|_0 - m_{0,i}} (1-\beta_0)^{m_{1,i}} \beta_0^{\|\delta_1\|_0 - m_{1,i}} \right\}
\end{aligned} \quad (15)$$

The formulation for the probability calculation is complete. However, performing this calculation directly requires a very high computational cost because there are many variables. Therefore, following the method used by (Zhang et al., 2022), we reduce the computational cost based on a recurrence relation.

We define $T(c, m)$ as follows and derive a recurrence relation.

$$\begin{aligned}
&T(c, m) \\
&= \sum_{\sum_{i=0}^{c}(m_{0,i}+m_{1,i})=m} \prod_{i=0}^{c} p(m_{0,i}, m_{1,i}) \\
&= \sum_{0 \le m_c \le \|\delta\|_0} \sum_{m_{0,c}+m_{1,c}=m_c} \sum_{\substack{\sum_{i=0}^{c-1}(m_{0,i}+m_{1,i}) \\ =m-m_c}} \prod_{i=0}^{c} p(m_{0,i}, m_{1,i}) \\
&= \sum_{0 \le m_c \le |\delta|} \left\{ \sum_{m_{0,c}+m_{1,c}=m_c} p(m_{0,c}, m_{1,c}) \times \right. \\
&\left. \sum_{\sum_{i=0}^{c-1}(m_{0,i}+m_{1,i})=m-m_c} \prod_{i=0}^{c-1} p(m_{0,i}, m_{1,i}) \right\} \\
&= \sum_{0 \le m_c \le \|\delta\|_0} T(0, m_c) \cdot T(c-1, m-m_c),
\end{aligned} \quad (16)$$

where

$$\begin{aligned}
p(m_0, m_1) = &\binom{\|\delta_0\|_0}{m_0} \binom{\|\delta_1\|_0}{m_1} \times \\
&\beta_0^{m_0} (1-\beta_0)^{\|\delta_0\|_0 - m_0} \beta_1^{m_1} (1-\beta_1)^{\|\delta_1\|_0 - m_1}
\end{aligned} \quad (17)$$

Then, we have

$$T(c, m) = \sum_{0 \le m_c \le \|\delta\|_0} T(0, m_c) \cdot T(c-1, m-m_c). \quad (18)$$

This allows us to compute any $T(c, m)$ from the recurrence relation in $O(e^2 \|\delta\|_0^2)$ time, once the initial values $T(0, m_c)$ for $0 \le m_c \le \|\delta\|_0$ are calculated. Since $P((X, D_\varepsilon) \in R(c, m))$ is a constant multiple of $T(c, m)$, we can also compute $P((X, D_\varepsilon) \in R(c, m))$ from the above. The calculation of $P((Y, \tilde{D_\varepsilon}) \in R(c, m))$ is the same as $P((X, D_\varepsilon) \in R(c, m))$.

Note that we need two noise size, $\|\delta_0\|_0$ and $\|\delta_1\|_0$, to calculate the probabilities although existing methods need only $\|\delta\|_0 = \|\delta_0\|_0 + \|\delta_1\|_0$. When calculating all combinations of $\|\delta_0\|_0$ and $\|\delta_1\|_0$ that satisfy $\|\delta\|_0 = \|\delta_0\|_0 + \|\delta_1\|_0$, the computational cost increases. Therefore, in the results of this paper, we first identify the worst-case $\|\delta_0\|$ and $\|\delta_1\|$ that satisfy $\|\delta\|_0 = \|\delta_0\|_0 + \|\delta_1\|_0$ and report on the results for this worst case. To be specific, for all $\|\delta_0\|_0$ and $\|\delta_1\|_0$ such that $\|\delta\|_0 = \|\delta_0\|_0 + \|\delta_1\|_0$, we compute C in Eq.(7) and adopt the $\|\delta_0\|_0$ and $\|\delta_1\|_0$ for which C is maximized as the worst-case noise.

4.4 Estimate $\underline{p_A}$ and $\overline{p_B}$

We estimate $\underline{p_A}$ and $\overline{p_B}$ according to the methodology described in (Jia et al., 2019). In our method, we obtain the output according to the formula, $l = g(x,D) = f(x \oplus \varepsilon, D \oplus \{\varepsilon_i | 1 \leq i \leq |D|\})$. For a specific label l, we assume that $P(g(x,D) = l) = p_l$. Furthermore, the operation of selecting a subsample D from D_{entire} and the generation of ε and ε_i are carried out independently. Therefore, $N_l = \sum_{i=1}^{N} \mathbb{I}[g(x,D) = l]$ follows a binomial distribution with the number of trials N and success probability p_l. From the above, the lower bound $\underline{p_A}$ for the probability p_A of outputting l_A and the upper bound $\overline{p_B}$ for the probability p_B of outputting l_B are estimated as

$$\begin{aligned} \underline{p_A} &= \text{Beta}\left(\frac{\alpha}{L}; N_{l_A}, N - N_{l_A} + 1\right), \\ \overline{p_B} &= \min\left(\text{Beta}\left(1 - \frac{\alpha}{L}; N_{l_B} + 1, N - N_{l_B}\right), 1 - \underline{p_A}\right) \end{aligned} \tag{19}$$

where α and L represent the significance level and the number of classes. Note that a Bonferroni correction is applied to the significance level in Eq.(19), which means dividing it by the number of test data $|D_{\text{test}}|$. By setting $\alpha = \frac{\alpha_{\text{entire}}}{|D_{\text{test}}|}$ for each test data, α_{entire} for the entire test dataset is achieved. We set $\alpha_{\text{entire}} = 5\%$ in our experiment.

5 EVALUATION

In this section, we show the results of the baseline and our method. The results of the baseline correspond to the existing method (Zhang et al., 2022) applied to a binary data classifier. Note that we do not apply the existing method as-is. In the evaluation of the baseline in this paper, the ensemble learning dataset is created using non-replacement sampling, and the probability calculations for deriving certified radius are focused solely on the locations where attack noise is introduced as described in section 4.3. This is because we conduct a fair comparison with the baseline and evaluate the effectiveness of the flexible noise.

In this experiment, we aim to address the following questions.

1. Is the flexible defensive noise of our method effective in improving the robustness certification?
2. What combinations of β_0 and β_1 in our method are suitable for each dataset?
3. How does the level of robustness vary if the poisoning size is changed?
4. How does the level of robustness vary if the attack noise size is changed?

5.1 Settings

Datasets. We evaluate the robustness of graph neural networks on MUTAG (Debnath et al., 1991), DHFR (Wale et al., 2008), NCI1 (Dobson and Doig, 2003) and AIDS (Riesen and Bunke, 2008) datasets. The MUTAG and DHFR datasets are split into training and test data with a ratio of 8:2. For the NCI1 and AIDS datasets, experiments are conducted with 250 training data and 50 test data for each label which are randomly chosen from the original datasets.

In our experiments, e, the size of the ensemble learning dataset D, is set to 50 for all datasets.

Models. We conduct experiments on graph convolutional networks (GCNs) (Kipf and Welling, 2016). Our model consists of two GCN layers and one linear classifier.

Metrics. We use certified accuracy (CA) for our evaluation. CA indicates the proportion of test data for which the prediction remains unchanged and the correct label is output, even if data are changed by the attacker. This metric is designed to evaluate both the performance and robustness of the model. In the certification of backdoor attacks, when evaluating CA, two axes can be considered. The first axis is the number of poisons p in the training dataset. In this case, the number of attack noise inserted into each poisoned data, $\|\delta\|_0$, is fixed at a constant value, and CA is evaluated while varying p. In particular, we describe CA at poisoning size p as CA_p. The second axis is the number of attack noise. In this case, p is fixed at a constant value and CA is evaluated while varying $\|\delta\|_0$. Specifically, we describe the CA at attack noise size $\|\delta\|_0$ as $CA_{\|\delta\|_0}$.

Then, CA_p and $CA_{\|\delta\|_0}$ are formulated as

$$\begin{aligned} CA_p &= \frac{\sum_{i=1}^{|D_{\text{test}}|} \mathbb{I}\left[R_x \geq p \ \wedge \ l = l^*\right]}{|D_{\text{test}}|} \quad \text{and} \\ CA_{\|\delta\|_0} &= \frac{\sum_{i=1}^{|D_{\text{test}}|} \mathbb{I}\left[R_x \geq \|\delta\|_0 \ \wedge \ l = l^*\right]}{|D_{\text{test}}|} \\ \text{where } l^* &= \underset{l \in \{1,\ldots,L\}}{\arg\max} \left(\sum_{i=1}^{N} \mathbb{I}[g(x,D) = l] \right), \end{aligned} \tag{20}$$

respectively. The range of p and $\|\delta\|_0$ are $[0, |D_{\text{entire}}|]$ and $[0, 100]$ in our experiments, respectively.

Explore the Optimal β_0 and β_1. We conduct the evaluation for large-size certification ($N = 1000$) after exploring the optimal value of the defensive noise through a heuristic approach using small-size certification ($N = 100$). In the existing method, evaluations

are conducted while varying the value of β. However, large-size certification requires a significant computational cost in our method, making it difficult to try various combinations of defensive noise probabilities. Therefore, we simplify the computation by optimizing the probability of defensive noise based on small-size certification in advance. We define the average certified radius, $\overline{R}$, for the test dataset as

$$\begin{aligned} \overline{R}_{\text{poison}} &= \sum_{p=0}^{p_{\max}} p \cdot (CA_p - CA_{p+1}), \\ \overline{R}_{\text{noise}} &= \sum_{\|\delta\|_0=0}^{\|\delta\|_0^{\max}} \|\delta\|_0 \cdot (CA_{\|\delta\|_0} - CA_{\|\delta\|_0+1}) \end{aligned} \tag{21}$$

for poisoning size and noise size certification, respectively. $CA_{p_{\max}+1}$ and $CA_{\|\delta\|_0^{\max}+1}$ are set to 0.

We report the results for large-size certification where the value of $\overline{R}_{\text{poison}}$ is maximized in small-size certification. In the small-size certification, we calculate the $\overline{R}_{\text{poison}}$ where $\|\delta\|_0 = 1$. In the certification of the baseline, we search for the optimal β in increments of 0.1 within the range of β = 0.6 to 0.9, satisfying $\beta_0 = \beta_1 = \beta$. In our method, we search for the optimal values of β_0 and β_1 separately in increments of 0.1 within the range of 0.2 to 0.9, satisfying $\beta_0 + \beta_1 > 1$. Note that when $\beta_0 + \beta_1 = 1$, data are completely randomized and $P(X = z \wedge D_\varepsilon = D_z) = P(Y = z \wedge \tilde{D}_\varepsilon = D_z)$. Thus, the model becomes completely unaffected by attack noise.

5.2 Experimental Results

Results for Poisoning Size. In this section, we report the results for poisoning size certification. This experiment confirms the model's robustness against backdoor attacks that inject p poison into the training dataset. We vary the poisoning size p in the range from 0 to $|D_{\text{entire}}|$ where the attack noise size $\|\delta\|_0$ is 1, 5, and 10. Fig. 2 and Fig. 3 shows the CA as a function of p in the baseline and our method, respectively. As can be seen from Fig. 2 and Fig. 3, on the all datasets, our method can maintain high CA against backdoor attacks with large poisoning size compared with the baseline.

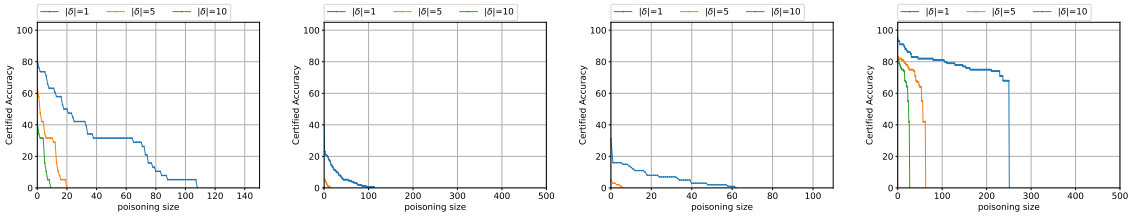
In the MUTAG dataset, when focusing on the case where $\|\delta\|_0 = 1$ (the blue line in Fig. 2(a)), the baseline demonstrates 63.15% CA when p is 10. In contrast, our method maintains a high CA, achieving 71.05% when the poisoning size is 10 as shown in Fig. 3(a). Furthermore, even in the other situations where noise sizes are 5 and 10, our method outperforms the baseline. For example, in the case where the noise size is 10 as shown in the green line in Fig. 2(a), the CA of the baseline is already 0 when the poisoning size is 20 . On the other hand, our method maintains approximately 20% CA as shown in Fig. 3(a). As a clearer indicator, $\overline{R}_{\text{poison}}$ shows that the baseline has $34.10, 5.57, 2.07$ and our method has $81.73, 20.34, 9.44$ on $\|\delta\|_0 = 1, 5, 10$, demonstrating the usefulness of the guarantees provided through flexible noise.

In the DHFR dataset, in the case where $\|\delta\|_0 = 1$ (the blue line in Fig. 2(b)), the baseline demonstrates 18.42% CA against poisoning attacks whose poisoning size is 10. In contrast, our method achieves 40.13% with the same poisoning size as shown in Fig. 3(b). Furthermore, our method retains more than 0% CA with larger poisoning sizes. Regarding the average certified radius, $\overline{R}_{\text{poison}}$ shows that the baseline has $7.87, 0.37, 0.00$ and our method has $71.38, 11.06, 4.05$ on $\|\delta\|_0 = 1, 5, 10$.

In the NCI1 dataset, the baseline demonstrates 13% CA when the poisoning size is 10, in the case where $\|\delta\|_0 = 1$ as shown in the blue line in Fig. 2(c). In contrast, our method achieves 16% CA at the same poisoning size as shown in Fig. 3(c). The difference in CA between our method and the baseline is relatively small compared with the results on other datasets. However, our method can maintain more than 0% CA even at larger poisoning sizes as with results on other datasets. Regarding the average certified radius, $\overline{R}_{\text{poison}}$ shows that the baseline has $4.55, 0.16, 0.00$ and our method has $8.85, 0.82, 0.05$ on $\|\delta\|_0 = 1, 5, 10$.

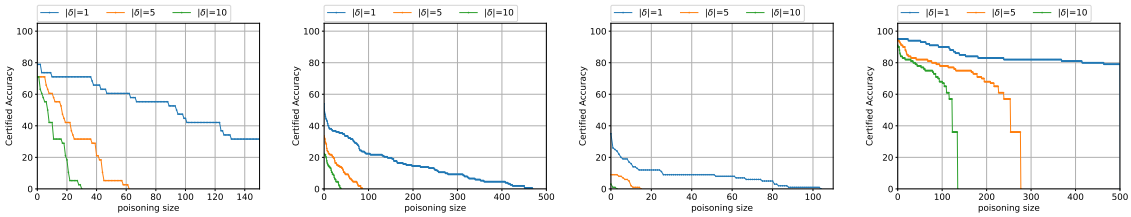
Finally, in the AIDS dataset, the baseline demonstrates 80% CA when the poisoning size is 100 as shown in the blue line in Fig. 2(d). In contrast, our method achieves approximately 90% CA when the poisoning size is 10 as shown in the blue line in Fig. 3(d). It is noteworthy that our method can keep approximately 80% CA even when the poisoning size is 500 whereas the CA of baseline is dropped to 0% when the size is 250. As for the other cases, our method maintains higher CA with larger poisoning sizes as shown in the orange and green lines in Fig. 3(d). Regarding the average certified radius, $\overline{R}_{\text{poison}}$ shows that the baseline has $198.08, 44.72, 18.88$ and our method has $423.33, 200.05, 96.52$ on $\|\delta\|_0 = 1, 5, 10$.

The experimental results confirm that, in general, applying our flexible noise significantly improves the model's robustness compared to adding defensive noise to each edge with the same probability. Additionally, for the MUTAG dataset, the model remains robust at high poisoning levels, when existing edges are removed with high probability. Therefore, if the defender would like to ensure correct predictions under large-scale poisoning attacks, defensive noise that deletes existing edges with high probability should be



(a) MUTAG ($\beta_0 = \beta_1 = 0.6$) (b) DHFR ($\beta_0 = \beta_1 = 0.6$) (c) NCI1 ($\beta_0 = \beta_1 = 0.6$) (d) AIDS ($\beta_0 = \beta_1 = 0.6$)

Figure 2: CA as a function of the poisoning size in the baseline.



(a) MUTAG ($\beta_0 = 0.8, \beta_1 = 0.3$) (b) DHFR ($\beta_0 = 0.3, \beta_1 = 0.8$) (c) NCI1 ($\beta_0 = 0.2, \beta_1 = 0.9$) (d) AIDS ($\beta_0 = 0.6, \beta_1 = 0.5$)

Figure 3: CA as a function of the poisoning size in our method.

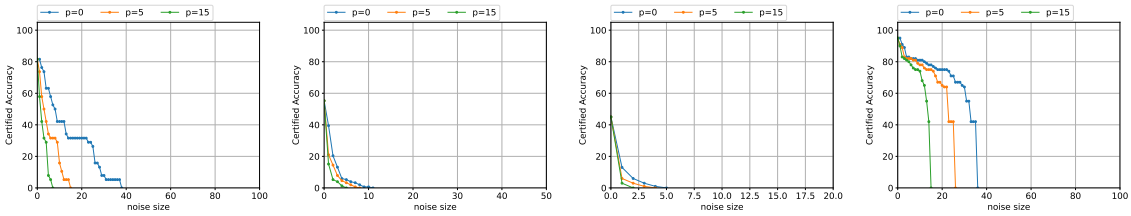
applied. On the other hand, for the DHFR and NCI1 datasets, defensive noise should focus on changing the elements of 0 in the adjacency matrix to 1 with high probability. Regarding the AIDS dataset, the optimal approach is to apply defense noise in a way that maintains more than half of the 0 and 1 edges. In this case, the defense noise is relatively similar to the baseline. However, it is remarkable that a 10% change in defensive noise probability results in such a significant improvement in robustness.

Results for Noise Size. We also report the results for the noise size certification in this section. This experiment confirms the model's robustness against backdoor attacks when $\|\delta\|_0$ edges of each graph data are changed. We vary the attack noise size $\|\delta\|_0$ in the range from 0 to 100. We evaluate the performance of the baseline and our method through varying the poisoning size p to 0, 5, and 15. Here, $p = 0$ is the same situation as evasion attacks. When evaluation is conducted while varying the noise size, the data size is normally required. However, in this paper, we intentionally disregard the data size in the evaluation. This is because our method can certify how the model is robust even when the attack range is larger than the size of data. For example, our method may compute a certified radius of 30 for data with a size of 25. In this case, we interpret it as meaning that our model's robustness has a surplus equivalent to 5 attack noises, and we directly reflect this in the CA evaluation. Fig. 4 and Fig. 5 shows the CA as a function of the noise size in the baseline and our method, respectively. As can be seen from Fig. 4 and Fig. 5, our method can retain higher CA on the all datasets with large noise sizes compared with the baseline.

In the MUTAG dataset, when focusing on the case where $p = 5$ (orange line in Fig. 4(a)), the baseline demonstrates 15.78% CA when noise is 10. On the othe hand, our method achieves 55.26% at the same noise size as shown in orange line in Fig. 5(a). Additionally, our method can maintain higher CA until the noise size exceeds 40 whereas the CA of the baseline is 0% before the noise size is 20. Even in the case where $p = 0$ and 10, the results demonstrate that our method is effective. Regarding the average certified radius, $\overline{R}_{\text{noise}}$ shows that the baseline has $11.86, 4.23, 1.73$ and our method has $36.39, 13.78, 5.86$ on $p = 0, 5, 15$.

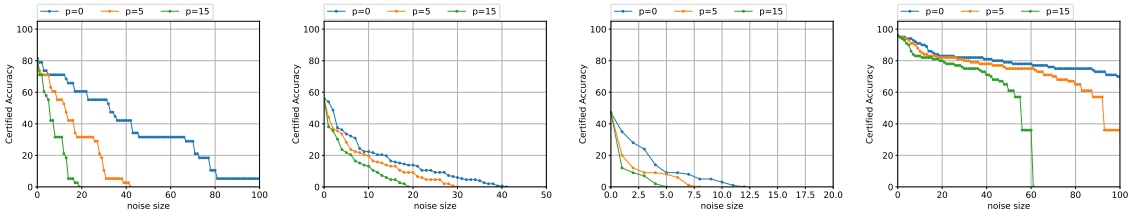
In the DHFR dataset, the CA of the baseline is 0% when the noise size exceeds 10 in the all cases. In particular, in the case where poisoning size is 15, the baseline demonstrates 0% CA when the noise size is 10 as shown in the green line in Fig. 4(b). In contrast, our method achieves 19.73% as shown in the green line in Fig. 5(b). Furthermore, our method retains higher CA with larger noise sizes compared with the baseline in all the cases. In this dataset, $\overline{R}_{\text{noise}}$ shows that the baseline has $0.94, 0.53, 0.25$ and our method has $6.36, 4.47, 2.75$ on $p = 0, 5, 15$.

In the NCI1 dataset, the baseline shows results similar to those on DHFR. For example, the baseline demonstrates 0% CA when the noise size is 5 as shown in the orange line in Fig. 4(c). In contrast, our method achieves 9.00% CA at the same noise size



(a) MUTAG ($\beta_0 = \beta_1 = 0.6$) (b) DHFR ($\beta_0 = \beta_1 = 0.6$) (c) NCI1 ($\beta_0 = \beta_1 = 0.6$) (d) AIDS ($\beta_0 = \beta_1 = 0.6$)

Figure 4: CA as a function of the noise size in the baseline ($\beta_0 = \beta_1 = 0.6$).



(a) MUTAG ($\beta_0 = 0.8, \beta_1 = 0.3$) (b) DHFR ($\beta_0 = 0.3, \beta_1 = 0.8$) (c) NCI1 ($\beta_0 = 0.2, \beta_1 = 0.9$) (d) AIDS ($\beta_0 = 0.6, \beta_1 = 0.5$)

Figure 5: CA as a function of the noise size in our method.

in the orange line in Fig. 5(c). In this dataset, $\overline{R}_{\text{noise}}$ shows that the baseline has $0.23, 0.10, 0.03$ and our method has $1.41, 0.65, 0.30$ on $p = 0, 5, 15$.

Finally, in the AIDS dataset, when we focus on the case where the poisoning size is 15, our method maintains 73.00% CA when the noise size is 40 as shown in the green line in Fig. 5(d). On the other hand, the CA of the baseline is already 0% at the same noise size as shown in the green line in Fig. 4(d). Regarding the average certified radius, $\overline{R}_{\text{noise}}$ shows that the baseline has $25.55, 18.05, 10.24$ and our method has $79.93, 72.74, 43.52$ on $p = 0, 5, 15$.

From the above results, it is confirmed that expanding the search range for defense noise also improves the $\overline{R}_{\text{noise}}$.

6 CONCLUSION

We have proposed a new robustness certification method against backdoor attacks in the graph domain by introducing flexible noise. As a result, the defender can explore a wider range of defensive noise parameters, allowing for more flexible handling against data modification attacks. Additionally, our method can use training datasets that include data of different sizes, providing a clear certification framework for classifiers that categorize a wide range of data types. In terms of computational complexity, our method is more efficient due to focusing calculations only on locations where attack noises are present. Our results demonstrate that adding flexible noise to binary elements is effective in improving the level of robustness certification.

Limitations and Future Work. However, our method has two limitations. First, our method is intended only for binary data classifiers among discrete data classifiers. Therefore, providing flexible robustness certification for other classifiers is a future challenge. Second, in this paper, we optimized the defensive noise based on the average certified radius. However, there are other possible approaches such as optimization based on CA with a certain poisoning or noise size. Therefore, it is also necessary to explore more practical optimization methods for use in robustness certification.

We hope that our flexible noise based robustness certification will inspire research on broader guarantees for discrete data classifiers in future.

REFERENCES

Chen, L., Li, J., Peng, J., Xie, T., Cao, Z., Xu, K., He, X., Zheng, Z., and Wu, B. (2020). A survey of adversarial learning on graphs. *arXiv preprint arXiv:2003.05730*.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., and Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molec-

ular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797.

Dobson, P. D. and Doig, A. J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783.

Feng, P., Ma, J., Li, T., Ma, X., Xi, N., and Lu, D. (2020). Android malware detection based on call graph via graph neural network. In *2020 International Conference on Networking and Network Applications (NaNA)*, pages 368–374. IEEE.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, T., Dolan-Gavitt, B., and Garg, S. (2019). Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Guo, L., Yin, H., Chen, T., Zhang, X., and Zheng, K. (2021). Hierarchical hyperedge embedding-based representation learning for group recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–27.

Jia, J., Cao, X., and Gong, N. Z. (2021). Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7961–7969.

Jia, J., Cao, X., Wang, B., and Gong, N. Z. (2019). Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. *arXiv preprint arXiv:1912.09899*.

Jiang, B. and Li, Z. (2022). Defending against backdoor attack on graph nerual network by explainability. *arXiv preprint arXiv:2209.02902*.

Jiang, C., He, Y., Chapman, R., and Wu, H. (2022). Camouflaged poisoning attack on graph neural networks. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 451–461.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kwon, H., Yoon, H., and Park, K.-W. (2019). Selective poisoning attack on deep neural network to induce fine-grained recognition error. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 136–139. IEEE.

Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., and Song, L. (2020). Heterogeneous graph neural networks for malicious account detection.

Meguro, R., Kato, H., Narisada, S., Hidano, S., Fukushima, K., Suganuma, T., and Hiji, M. (2024). Gradient-based clean label backdoor attack to graph neural networks. In *ICISSP*, pages 510–521.

Qiu, R., Huang, Z., Li, J., and Yin, H. (2020). Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–23.

Riesen, K. and Bunke, H. (2008). Iam graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings*, pages 287–297. Springer.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

Wale, N., Watson, I. A., and Karypis, G. (2008). Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375.

Wang, B., Jia, J., Cao, X., and Gong, N. Z. (2021). Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1645–1653.

Wang, S., Chen, Z., Ni, J., Yu, X., Li, Z., Chen, H., and Yu, P. S. (2019). Adversarial defense framework for graph neural network. *arXiv preprint arXiv:1905.03679*.

Weber, M., Xu, X., Karlaš, B., Zhang, C., and Li, B. (2023). Rab: Provable robustness against backdoor attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1311–1328. IEEE.

Yang, J., Ma, W., Zhang, M., Zhou, X., Liu, Y., and Ma, S. (2021). Legalgnn: Legal information enhanced graph neural network for recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–29.

Zhang, M., Hu, L., Shi, C., and Wang, X. (2020). Adversarial label-flipping attack and defense for graph neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 791–800. IEEE.

Zhang, X. and Zitnik, M. (2020). Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275.

Zhang, Y., Albarghouthi, A., and D'Antoni, L. (2022). Bagflip: A certified defense against data poisoning. *Advances in Neural Information Processing Systems*, 35:31474–31483.

Zhang, Z., Jia, J., Wang, B., and Gong, N. Z. (2021). Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pages 15–26.

Zügner, D., Akbarnejad, A., and Günnemann, S. (2018). Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2847–2856.