A Details of Related Works

A.1 Graph Contrastive Learning

GCL has recently gained significant attention and shows promise for improving graph representations in scenarios where labeled data is scarce [31, 32, 9, 30, 27, 15, 33, 32]. Generally, GCL creates two views through data augmentation and contrasts representations between two views. GraphCL [9] focuses on graph classification by exploring four types of augmentations including node dropping, edge perturbation, attribute masking and subgraph sampling. GRACE [30] and GCA [27] adapt SimCLR [55] to graphs to maximize the mutual information between two views for each node through a variety of data augmentation. DGI [31] applied InfoMax principle [56] to train a GNN encoder by maximizing the mutual information between node-level and graph-level representations. MVGRL [32] proposes to learn representations by maximizing the mutual information between the cross-view representations of nodes and graphs. Several recent works [16, 36] show that GCL is vulnerable to adversarial attacks. CLGA [16] is an unsupervised poisoning attacks for attack graph contrastive learning. In detail, the gradients of the adjacency matrices for both views are computed, and edge flipping is performed using gradient ascent to maximize the contrastive loss. Despite very few empirical works [9, 15, 14] on robustness of GCL, there are no existing works studying the certified robustness of GCL. In contrast, we propose to provide robustness certificates for GCL by using randomized edgedrop smoothing. To the best of our knowledge, our method is the first work to study the certified robustness of GCL.

A.2 Certifiable Robustness of Graph Neural Networks

Several recent studies investigate the certified robustness of GNNs in the supervised setting [23, 28, 29, 26, 24, 25, 37]. Zügner et al. [23] are the first to explore certifiable robustness with respect to node feature perturbations. Subsequent works[28, 26, 29, 24, 38] extend the analysis to certifiable robustness under topological attacks. For instance, Bojchevski et al. [28] propose a branch-and-bound algorithm to achieve tight bounds on the global optimum of certificates for topological attacks. Bojchevski et al. [29] further adapt the randomized smoothing technique to sparse settings, deriving certified robustness for GNNs. This approach involves injecting random noise into test samples to mitigate the negative effects of adversarial perturbations. Wang et al. [24] further refine this technique to provide theoretically tight robust certificates. Our work is inherently different from them: (**i**) existing work focues on the certified robustness of GNN under (semi)-supervised setting; while we provide a unified definition to evaluate and certify the robustness of GCL for unsupervised representation learning. (**ii**) we theoretically provide the certified robustness for GCL in the absence of labeled data, and this certified robustness can provably sustained in downstream tasks. (**iii**) we design an effective training method to enhance the robustness of GCL.

B Preliminary of Randomized Smoothing

One potential solution for achieving certified robustness in GNNs is binary randomized smoothing presented in [24]. Specifically, consider a noisy vector ϵ in the discrete space $\{0,1\}^N$

$$\mathbb{P}(\epsilon_i = 0) = \beta, \quad \mathbb{P}(\epsilon_i = 1) = 1 - \beta, \tag{B.1}$$

where $i = 1, 2, \cdots, N$. This indicates that the connection status of the i-th entry of $\mathbf{v}$ will be flipped with probability $1 - \beta$ and preserved with probability β. Given a base node or graph classifier $f(\mathbf{v})$ that returns the class label with highest label probability, the smoothed classifier g is defined as:

$$g(\mathbf{v}) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(f(\mathbf{v} \oplus \epsilon) = y), \tag{B.2}$$

where $\mathcal{Y}$ denotes the label set, and $\oplus$ represents the XOR operation between two binary variables, which combines the structural information of both variables. $\mathbb{P}(f(\mathbf{v} \oplus \epsilon) = y)$ is the probability that the base classifier f predicts class y when random noise ϵ is added to $\mathbf{v}$. According to [24], by injecting multiple ϵ to $\mathbf{v}$ and returning the most likely class y_A (the majority vote), if it is holds that:

$$\min \mathbb{P}(f(\mathbf{v} \oplus \epsilon) = y_A) \geq \max_{y \neq y_A} \mathbb{P}(f(\mathbf{v} \oplus \epsilon) = y), \text{ s.t. } \|\epsilon\|_0 \leq k, \tag{B.3}$$

where y_A is the class with the highest label probability. Then we can guarantee there exists a certified perturbation size k such that for any perturbation δ with $||\delta||_0 \leq k$, the predictions of $\mathbf{v} \oplus \delta$ will

remain unchanged, that is, $g(\mathbf{v} \oplus \delta) = f(\mathbf{v} \oplus \delta \oplus \epsilon) = f(\mathbf{v} \oplus \epsilon) = y_A$. Thus, f is certifiably robust at $(\mathbf{v}, y_A)$ when perturbing at most k edges. For more deatils of the proof, please refer to [24].

However, directly applying the vanilla randomized smoothing approach to certify robustness in GCL is inapplicable. This approach may require injecting random noise into the entire graph during data augmentation, resulting in an excessive number of noisy/spurious edges that can significantly harm downstream task performance, particularly for large and sparse graphs. In contrast, our proposed method, Randomized Edgedrop Smoothing (RES), aims to prevent the introduction of excessive spurious edges in graphs. RES achieves this by injecting randomized edgedrop noises, where observed edges are randomly removed from the graphs with a certain probability. Additionally, we propose an effective training method for robust GCL. Our experimental results in Sec. 6.4 demonstrate that our method significantly outperforms the ablative method, FLIP, that directly applies vanilla randomized smoothing [24] in GCL. Specifically, on the clean Cora and Pubmed graphs, our method achieves accuracies of 82% and 85.6% respectively. On the Nettack perturbed [11] Cora and Pubmed graphs, our method achieves robust accuracies of 79.4% and 82.5%, respectively. In contrast, FLIP achieves only up to 27% and 42% robust accuracies on the Cora and Pubmed datasets, respectively. These results strongly validate the effectiveness of our RES approach,

C Detailed Proofs

C.1 Proof of Theorem 1

To begin, we provide a formalization of *similarity* in relation to the latent class.

Given a GNN encoder h and an input sample $\mathbf{v}$, which consists of a positive sample $\mathbf{v}^+$ and a negative sample $\mathbf{v}^-$ an input sample $\mathbf{v}$ with positive sample $\mathbf{v}^+$ and negative sample $\mathbf{v}^-$, according to Def. 1, we assume that $c^+, c^- \in \mathcal{C}$ represent the latent classes of $\mathbf{v}^+$ and $\mathbf{v}^-$ in the latent space of h, respectively. These latent classes are based on the distribution η over $\mathcal{C}$. we assume that $c^+, c^- \in \mathcal{C}$ are the latent class of $\mathbf{v}^+$ and $\mathbf{v}^-$ in the latent space of h, respectively, which are randomly determined based on the distribution η on $\mathcal{C}$. Similar data $h(\mathbf{v}), h(\mathbf{v}^+)$ are i.i.d. draws from the same class distribution $\mathcal{D}_{c^+}$, whereas negative samples originate from the marginal of $\mathcal{D}_{sim}$, which are formalized as follow:

$$\begin{aligned} \mathcal{D}_{sim}(\mathbf{v}, \mathbf{v}^+) &= \mathop{\mathbb{E}}_{c^+ \sim \eta} \mathcal{D}_{c^+}(\mathbf{v}) \mathcal{D}_{c^+}(\mathbf{v}^+). \\ \mathcal{D}_{neg}(\mathbf{v}^-) &= \mathop{\mathbb{E}}_{c^- \sim \eta} \mathcal{D}_{c^-}(\mathbf{v}^-). \end{aligned} \tag{C.4}$$

Since classes are allowed to overlap and/or be fine-grained [39], this is a plausible formalization of "similarity". The formalization connects similarity with latent class, thereby offering a viable way to employ the similarity between samples under the latent space h for defining the probability of $\mathbf{v}$ being the positive sample of $\mathbf{v}^+$.

Lemma 1 (The Generalized Extreme Value Distribution [44])**.** Let $\mathbf{v}_1, \mathbf{v}_2, \cdots$ be a sequence of i.i.d samples from a common distribution function F. Let $M_N = \max\{\mathbf{v}_1, \cdots, \mathbf{v}_N\}$, indicting the maximum of $\{\mathbf{v}_1, \cdots, \mathbf{v}_N\}$. Then if $\{a_N > 0\}$ and $\{b_N \in \mathbb{R}\}$ are sequences of constants such that

$$\mathbb{P}\{(M_N - b_N)/a_N \leq z\} \to G(z), \tag{C.5}$$

where G is a non-degenerate distribution function. It is a member of the generalized extreme value family of distributions, which belongs to either the Gumbel family (Type I), the Fréchet family (Type II) or the Reverse Weibull family (Type III) with their CDFs as follows:

$$\begin{aligned} &\text{Gumbel family (Type I):} \quad G(z) = \exp\{-\exp[-(\frac{z-b}{a})]\}, \quad z \in \mathbb{R}, \\ &\text{Fréchet family (Type II):} \quad G(z) = \begin{cases} 0, & \text{z<b}, \\ \exp\{-(\frac{z-b}{a})^{-\sigma}\}, & \text{z}\geq\text{b}, \end{cases} \\ &\text{Reverse Weibull family (Type III):} \quad G(z) = \begin{cases} \exp\{-(\frac{b-z}{a})^{-\sigma}\}, & \text{z<b}, \\ 1, & \text{z}\geq\text{b}, \end{cases} \end{aligned} \tag{C.6}$$

where $a > 0$, $b \in R$ and $\sigma > 0$ are the scale, location and shape parameters, respectively.

Theorem 1. *Let $\mathbb{B}(\mathbf{v}^+)$ be a space around $\mathbf{v}^+$ as defined in Eq. 4. Given an input sample $\mathbf{v}$ and a GNN encoder h learnt by GCL in Eq. 1, the probability of $\mathbf{v}$ being the positive sample of $\mathbf{v}^+$ is:*

$$Pr(\mathbf{v} \in \mathbb{B}(\mathbf{v}^+); h) = \exp\big[-(\frac{1 - s(h(\mathbf{v}), h(\mathbf{v}^+))}{a})^{\sigma}\big], \tag{C.7}$$

where $s(\cdot,\cdot)$ is the cosine similarity function. a, $\sigma > 0$ are Weibull shape and scale parameters [44].

Proof. We first assume $\{h(\mathbf{v}_1), h(\mathbf{v}_2), \cdots\}$ are i.i.d samples under the latent space h. Then, suppose $\mathbf{V}^- = \{\mathbf{v}_1^-, \cdot, \mathbf{v}_n^-\}$ is the set of negative samples of $\mathbf{v}$. We assume there exists a continuous non-degenerate margin distribution M, which is denoted as follows:

$$M := \min_{i \in [n]} D_i, \quad \text{with } D_i := (1 - s(h(\mathbf{v}^+), h(\mathbf{v}_i^-)))/2, \tag{C.8}$$

where $M, D_i \in [0, 1]$, and $s(\cdot,\cdot)$ is the cosine similarity function. M indicates the half of the minimum distance between $h(\mathbf{v}^+)$ and $h(\mathbf{v}_i^-)$.

According to Lemma 1, we know that there exists $G(z)$ in the three types of generalized extreme value family. And since Lemma 1 is applied to the maximum, refer to Eq. (C.8), we transfer the variable M to $\overline{M} := \max_{i \in [n]} -D_i$, Since $-D_i$ is bounded ($-D_i < 0$), let $b = 0$, the marginal distribution of D_i series, for $i = 1, 2, \cdots$, can be Reverse Weibull family. Therefore, the asymptotic marginal distribution of $\overline{M}$ fits into the Reverse Weibull distribution:

$$G(z) = \begin{cases} \exp\{-(\frac{-z}{a})^{-\sigma}\}, & \text{z<0,} \\ 1, & \text{z}\geq\text{0,} \end{cases}$$

where $\sigma > 0$ is the shape parameter and a is the scale parameter. Compared to Eq. (C.6), here, $b = 0$ because $-D_i$ is bound ($-D_i < 0$). We use margin distances D_i of the λ closest samples with $\mathbf{v}^+$ to estimate the parameters α and σ, which means to estimate $\widehat{W}$ of the distribution function W.

Following Eq. (C.4), if the similarity $s(h(\mathbf{v}), h(\mathbf{v}^+))$ is larger, $\mathbf{v}$ is more possible to be from the same latent class as $\mathbf{v}^+$, which implies $\mathbf{v}$ is the positive sample of $\mathbf{v}^+$. Since the distance between $h(\mathbf{v})$ and $h(\mathbf{v}^+)$ can be denoted as $1 - s(h(\mathbf{v}^+), h(\mathbf{v}_i^-))$, the probability of $\mathbf{v}$ is the positive sample of $\mathbf{v}^+$ can be written as:

$$\begin{aligned} &\mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}^+); h) \\ =&\mathbb{P}(1 - s(h(\mathbf{v}), h(\mathbf{v}^+)) < \min\{D_1, \cdots, D_n\}) \\ =&\mathbb{P}(s(h(\mathbf{v}), h(\mathbf{v}^+) - 1) > \max\{-D_1, \cdots, -D_n\}) \\ =&\mathbb{P}(\overline{M} < s(h(\mathbf{v}), h(\mathbf{v}^+)) - 1) \\ =&\widehat{W}(s(h(\mathbf{v}), h(\mathbf{v}^+)) - 1). \end{aligned} \tag{C.9}$$

Let $b_n = 0$, $a_n = 0$ and $z = s(h(\mathbf{v}), h(\mathbf{v}^+)) - 1$, since $z = s(h(\mathbf{v}), h(\mathbf{v}^+)) - 1 < 0$, we can rewrite Eq. (C.9) as (C.7) according to Lemma 1. □

C.2 Proof of Theorem 2

For simplicity of notation, let $p_{\mathbf{v}^+,h}(\mathbf{v}) = \mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}^+); h)$. Then, given a randomized edgedrop noise $\epsilon \in \mathcal{D}_\epsilon$ with the following probability distribution:

$$\mathbb{P}(\epsilon_i = 0 | \mathbf{v}_i = 0) = 1,\ \mathbb{P}(\epsilon_i = 0 | \mathbf{v}_i = 1) = \beta,\ \text{and}\ \mathbb{P}(\epsilon_i = 1 | \mathbf{v}_i = 1) = 1 - \beta, \tag{C.10}$$

where ϵ_i is the i-entry of noisy vector ϵ and $\mathbf{v}_i$ is the connection status of i-th entry of $\mathbf{v}$. Then, we let

$$\underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon) = \inf_{\epsilon \in \mathcal{D}_\epsilon} p_{\mathbf{v}^+,h}(\mathbf{v} \oplus \epsilon),\ \ \overline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon) = \sup_{\epsilon \in \mathcal{D}_\epsilon} p_{\mathbf{v}^+,h}(\mathbf{v} \oplus \epsilon), \tag{C.11}$$

where $\underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon)$ and $\overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon)$ denote the lower bound and the upper bound on $p_{\mathbf{v}^+,h}(\mathbf{v} \oplus \epsilon)$, respectively. The formal theorem is presented as follows:

Theorem 2. *Let $\mathbf{v}$ be a clean input and $\mathbf{v}' = \mathbf{v} \oplus \delta$ be its perturbed version, where $||\delta||_0 \leq k$. $\mathbf{V}^- = \{\mathbf{v}_1^-, \cdots, \mathbf{v}_n^-\}$ is the set of negative samples of $\mathbf{v}$. If for all $\mathbf{v}_i^- \in \mathbf{V}^-$:*

$$\underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon) - \max_{\mathbf{v}_i^- \in \mathbf{V}^-} \overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon) > 2\Delta, \tag{C.12}$$

where

$$\Delta = 1 - \frac{\binom{d}{e}}{\binom{d+k}{e}} \cdot \beta^k, \tag{C.13}$$

and $e = \|\mathbf{v} \oplus \epsilon\|_0$ denotes the number of remaining edges of $\mathbf{v}$ after injecting ϵ, then with a confidence level of at least $1 - \alpha$, we have:

$$p_{\mathbf{v}^+,h}(\mathbf{v}' \oplus \epsilon) > \max_{\mathbf{v}_i^- \in \mathbf{V}^-} p_{\mathbf{v}_i^-,h}(\mathbf{v}' \oplus \epsilon). \tag{C.14}$$

Proof. Suppose $\mathbf{V} = \{\mathbf{v}^+, \mathbf{v}_1^-, \cdots, \mathbf{v}_n^-\}$ is the set of positive and negative samples of $\mathbf{v}$. For any $\mathbf{v}_i \in \mathbf{V}$, let $p_{\mathbf{v}_i,h}(\mathbf{v}) = \mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}_i); h)$ for simplicity, we have:

$$\begin{aligned} p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) &= \mathbb{P}(\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h) \\ p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) &= \mathbb{P}(\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h). \end{aligned} \tag{C.15}$$

As stated by the law of total probability, we have:

$$\begin{aligned} p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) = \mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]) \\ + \mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]) \\ p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) = \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]) \\ + \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]), \end{aligned} \tag{C.16}$$

where $(\mathbf{v}' \oplus \epsilon) \cap \delta$ represent the intersection of the edge sets $(\mathbf{v}' \oplus \epsilon)$ and δ, that is, the set of edges shared in the two structure vectors. Therefore, for $\mathbf{v}' \oplus \epsilon \cap \delta = \emptyset$, it means that no noisy edge of δ exists in $(\mathbf{v}' \oplus \epsilon)$, which indicates that $\mathbf{v} \oplus \epsilon$ and $\mathbf{v}' \oplus \epsilon$ are structural identical at all indices, i.e., $\mathbf{v}' \oplus \epsilon = \mathbf{v} \oplus \epsilon$. Therefore, we can then derive the following equality:

$$\mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \mid [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]) = \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \mid [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]). \tag{C.17}$$

By multiplying $\mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset)$ in both sides of Eq. (C.17), we have:

$$\mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]) = \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset]). \tag{C.18}$$

Combining Eq. (C.16) with Eq. (C.18) results in:

$$\begin{aligned} p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) - p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) =& \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]) - \\ & \mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]). \end{aligned} \tag{C.19}$$

Since probabilities are non-negative, we can rewrite Eq. (C.19) into the following inequality:

$$\begin{aligned} & p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) - \mathbb{P}([\mathbf{v} \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]) \\ & \leq p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) \leq \\ & p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) + \mathbb{P}([\mathbf{v}' \oplus \epsilon \in \mathbb{B}(\mathbf{v}_i); h] \wedge [(\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset]). \end{aligned} \tag{C.20}$$

Applying the conjunction rule, we have:

$$\begin{aligned} & p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) - \mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset) \leq p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) \leq \\ & p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) + \mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset). \end{aligned} \tag{C.21}$$

Since $\mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset) = 1 - \mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset)$, and $(\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset$ implies that remaining edges of $\mathbf{v} \oplus \epsilon$ and $\mathbf{v}' \oplus \epsilon$ are identical after injecting the random masking noise, we can derive the following probability:

$$\mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta = \emptyset) = \frac{\binom{d}{e}}{\binom{d+|\delta|}{e}} \cdot \beta^{|\delta|}, \tag{C.22}$$

where e denotes the number of remaining edges, and $k = |\delta|$ denotes the certified perturbation size. $\beta^e(1-\beta)^{d-e+|\delta|}$ represents the probability that $|\delta|$ noise edges are all dropped and e edges of $\mathbf{v}$ are retained in $\mathbf{v}' \oplus \epsilon$. Therefore, we have:

$$\mathbb{P}((\mathbf{v}' \oplus \epsilon) \cap \delta \neq \emptyset) = 1 - \frac{\binom{d}{e}}{\binom{d+|\delta|}{e}} \cdot \beta^{|\delta|} \leq 1 - \frac{\binom{d}{e}}{\binom{d+k}{e}} \cdot \beta^k = \Delta. \tag{C.23}$$

Substituting Eq. (C.23) into Eq. (C.21), then Eq. (C.21) can be rewritten as

$$p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) - \Delta \leq p_{\mathbf{v}_i,h}(\mathbf{v}' \oplus \epsilon) \leq p_{\mathbf{v}_i,h}(\mathbf{v} \oplus \epsilon) + \Delta. \tag{C.24}$$

Referring to Eq. (C.24), for any $\mathbf{v}^+$ and corresponding $\mathbf{v}_i^- \in \mathbf{V}^-$ we have

$$\begin{aligned} p_{\mathbf{v}^+,h}(\mathbf{v}' \oplus \epsilon) &\geq p_{\mathbf{v}^+,h}(\mathbf{v} \oplus \epsilon) - \Delta, \\ p_{\mathbf{v}_i^-,h}(\mathbf{v} \oplus \epsilon) + \Delta &\geq p_{\mathbf{v}_i^-,h}(\mathbf{v}' \oplus \epsilon). \end{aligned} \tag{C.25}$$

Thus, we can derive the following inequality based on Eq. (C.25):

$$\begin{aligned} &p_{\mathbf{v}^+,h}(\mathbf{v}' \oplus \epsilon) \geq p_{\mathbf{v}^+,h}(\mathbf{v} \oplus \epsilon) - \Delta \geq \underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon) - \Delta \\ \geq &\max_{\mathbf{v}_i^- \in \mathbf{V}} \overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon) + \Delta \geq \overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon) + \Delta \geq p_{\mathbf{v}_i^-,h}(\mathbf{v} \oplus \epsilon) + \Delta \\ \geq &p_{\mathbf{v}_i^-,h}(\mathbf{v}' \oplus \epsilon), \end{aligned} \tag{C.26}$$

which can be restated as Eq. (C.14). This completes our proof. □

C.3 Proof of Theorem 3

In order to transfer the certified robustness of GCL to downstream tasks, we first introduce two loss functions, namely, unsupervised loss and supervised loss. Subsequently, we introduce a lemma to establish the relationship between GCL and downstream tasks. Finally, a theorem is proposed to prove that the certified robustness of GCL is provably preserved in downstream tasks.

Unsupervised Loss Given an input sample $\mathbf{v}$ with its positive sample $\mathbf{v}^+$ and n negative samples $\{\mathbf{v}_1^-, \ldots, \mathbf{v}_n^-\}$. Let $h : \{0,1\}^n \to \mathbb{R}^d$ be the GNN encoder based on GCL to obtain representations. The unsupervised loss for h at point $\mathbf{v}$ is defined as:

$$L_{un}(\mathbf{v}; h) := \sum_{i=1}^{n} \ell(h(\mathbf{v})^\top (h(\mathbf{v}^+) - h(\mathbf{v}_i^-))), \tag{C.27}$$

where l is logistic loss $l(\mathbf{v}) = \log\left(1 + \sum_{i=1}^{n} \exp\left(-\mathbf{v}_i\right)\right)$ according to [57, 39]. Note that this loss is essentially equivalent to the InfoNCE loss [42, 57] shown in Eq. 1, which is widely-used for GCL.

Supervised Loss Linear evaluation, which learns a downstream linear layer after the base encoder, is a common way to evaluate the performance of GCL model in downstream tasks. Let $\mathcal{C}$ be denoted as the set of latent classes, where $|\mathcal{C}| = m$. We consider the standard supervised learning tasks that classify a data point into one of the classes in $\mathcal{C}$. To connect the GCL task with the downstream classification task, the supervised loss of downstream classifier f at $(\mathbf{x}, y)$ is defined as:

$$L_{sup}(\mathbf{x}, y; f) := \ell(\{f(x)_y - f(x)_{y'}\}_{y' \neq y}), \tag{C.28}$$

where l is the same as the loss function used in the unsupervised loss in Eq. (C.27). To evaluate the learned representations on downstream tasks, we typically fix h and train a linear classifier $\mathbf{W} \in \mathbb{R}^{m \times d}$ on the top of the encoder h. Therefore, the supervised loss of h at at $(\mathbf{x}, y)$ is defined as:

$$L_{sup}(\mathbf{x}, y; h) := \inf_{\mathbf{W} \in \mathbb{R}^{m \times d}} L_{sup}(\mathbf{x}, y; \mathbf{W}h), \tag{C.29}$$

Lemma 2 (Connection between GCL and Downstream Tasks [39])**.** Given a input sample $\mathbf{v}$ with its positive sample $\mathbf{v}^+$ and the set of negative samples $\mathbf{V}^- = \{\mathbf{v}_1^-, \ldots, \mathbf{v}_n^-\}$. $\mathcal{C}$ is the set of latent class with distribution η. The latent class of $\mathbf{v}^+$ is denoted as $c^+ \in \mathcal{C}$. Suppose any $\mathbf{v}^- \in \mathbf{V}^-$ has the latent class $c^- \in \mathcal{C}$. Then we have

$$L_{un}(\mathbf{v}; h) \geq (1 - \tau) L_{sup}(\mathbf{v}, c^+; h) + \tau, \tag{C.30}$$

where $\tau = \mathop{\mathbb{E}}_{c^+, c^- \sim \eta} \mathbf{1}\{c^+ = c^-\}$, which indicates the expectation that $c^+ = c^-$.

Proof. Based on the definitions of $L_{un}(\mathbf{v}; h)$ and $L_{sup}(\mathbf{v}; h)$ in Eq. (C.27) and Eq. (C.29), we have

$$\begin{aligned} L_{un}(\mathbf{v}; h) &= \mathop{\mathbb{E}}_{c^+, c^- \sim \eta^2} [\mathop{\mathbb{E}}_{\mathbf{v}^+ \sim \mathcal{D}_{c^+}, \mathbf{v}^- \sim \mathcal{D}_{c^-}} \ell(h(\mathbf{v})^\top (h(\mathbf{v}^+) - h(\mathbf{v}^-)))] \\ &\geq \mathop{\mathbb{E}}_{c^+, c^- \sim \eta^2} \ell(h(\mathbf{v})^\top (\mathop{\mathbb{E}}_{\mathbf{v}^+ \sim \mathcal{D}_{c^+}} [h(\mathbf{v}^+)] - \mathop{\mathbb{E}}_{\mathbf{v}^- \sim \mathcal{D}_{c^-}} [h(\mathbf{v}^-)])) \\ &= (1 - \tau) \mathop{\mathbb{E}}_{c^+, c^- \sim \eta^2} [L_{sup}(\mathbf{v}, c^+; f) | c^+ \neq c^-] + \tau \\ &= (1 - \tau) L_{sup}(\mathbf{v}, c^+; h) + \tau. \end{aligned} \tag{C.31}$$

This completes proof. □

Note that the above bound is similar to Lemma 4.3 in [39]. By leveraging our Theorem 2, we can establish the connection between GCL and downstream tasks, and use this connection to prove the transferability of the certified robustness of GCL to downstream tasks.

Theorem 3. *Given a GNN encoder h trained via GCL and an clean input $\mathbf{v}$. $\mathbf{v}^+$ and $\mathbf{V}^- = \{\mathbf{v}_1^-, \cdots, \mathbf{v}_n^-\}$ are the positive sample and the set of negative samples of $\mathbf{v}$, respectively. Let c^+ and c_i^- denote the latent classes of $\mathbf{v}^+$ and $\mathbf{v}_i^-$, respectively. Suppose f is the downstream classifier that classify a data point into one of the classes in $\mathcal{C}$. Then, we have*

$$\mathbb{P}(f(h(\mathbf{v})) = c^+) > \max_{\mathbf{v}_i^- \in \mathbf{V}^-} \mathbb{P}(f(h(\mathbf{v})) = c_i^-) \tag{C.32}$$

Proof. Since $\mathbf{v}$ is a clean input, according to Eq. (C.4) and Sec. 4.1, the positive pair $(\mathbf{v}, \mathbf{v}^+)$ a pair of similar data that come from the same class distribution $\mathcal{D}_{c^+}$ and they have the following relationship:

$$\mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}^+); h) > \max_i \mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}_i^-); h). \tag{C.33}$$

Then, based on Theorem 1, we know that $\mathbb{P}(\mathbf{v} \in \mathbb{B}(\mathbf{v}^+); h)$ is monotonically increasing as $s(h(\mathbf{v}), h(\mathbf{v}^+))$ increases. Therefore, we have

$$s(h(\mathbf{v}), h(\mathbf{v}^+)) > \max_i s(h(\mathbf{v}), h(\mathbf{v}_i^-)) \tag{C.34}$$

According to Eq. (C.34), we can obtain that:

$$\sum_{i=1}^{n} [s(h(\mathbf{v}), h(\mathbf{v}^+)) - s(h(\mathbf{v}), h(\mathbf{v}_i^-))] > 0, \tag{C.35}$$

and the equivalent form of Eq. (C.35) is given as:

$$(\tilde{s}(h(\mathbf{v}^+)) - \tilde{s}(h(\mathbf{v}^-)))^\top h(\mathbf{v}) > 0, \forall \mathbf{v}^- \in \{\mathbf{v}_1^-, \cdots, \mathbf{v}_n^-\}, \tag{C.36}$$

where $\tilde{s}$ is the l_2-normalization operation on vector $\mathbf{v}$. Then, we can obtain:

$$(\tilde{s}(h(\mathbf{v}^+)) - \tilde{s}(h(\mathbf{v}^-)))^\top \tilde{s}(h(\mathbf{v})) > 0, \forall \mathbf{v}^- \in \{\mathbf{v}_1^-, \cdots, \mathbf{v}_n^-\}. \tag{C.37}$$

To relate the robustness of GCL to that of downstream tasks, we select the negative samples whose latent class c_i^- is different from c^+ and obtain the following relationship:

$$\sum_{i=1}^{n} [(\tilde{s}(h(\mathbf{v}^+)) - \tilde{s}(h(\mathbf{v}^-)))^\top \tilde{s}(h(\mathbf{v})) | c^+ \neq c_i^-] > 0. \tag{C.38}$$

Substitute the left side of Eq. (C.38) into Eq (C.27), we have $L_{un}(\mathbf{v}; h) < 1$. According to Lemma 2, we know that:

$$L_{sup}(\mathbf{v}, c^+; h) \leq \frac{L_{un}(\mathbf{v}; h) - \tau}{1 - \tau} < 1. \tag{C.39}$$

Hence, according to the definition of L_{sup} in Eq. (C.29), we have:

$$\mathbb{P}(f(h(\mathbf{v})) = c^+) > \mathbb{P}(f(h(\mathbf{v})) = c^-), \ \forall c^- \in \{c_1^-, \cdots, c_n^-\} \tag{C.40}$$

which means that the logit output of the positive latent class c^+ is always larger than any negative latent class c_i^- for $\mathbf{v}$. Thus, we can rewrite Eq. (C.40) to Eq. (C.32), and conclude the proof. □

Algorithm 1 The Training Algorithm of RES.

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$.
Output: Trained GNN encoder h_θ.
1: Randomly initialize θ for h_θ;
2: **for** epoch=$1, 2, \ldots,$ **do**
3: Generate two augmented graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ by $q_i(\mathcal{G}) \sim \mathcal{T}$ and $q_j(\mathcal{G}) \sim \mathcal{T}$;
4: Inject randomized edgedrop noise ϵ to $\mathcal{G}_i$;
5: Obtain node or graph representation $\mathbf{Z}_i$ and $\mathbf{Z}_j$ from $\mathcal{G}_i$ and $\mathcal{G}_j$ by using h_θ;
6: Update θ by applying gradient descent to minimize Eq. (1).
7: **end for**
8: **return** h_θ;

By applying Theorem 3, we can demonstrate that given $\mathbf{v}$'s perturbed version $\mathbf{v}' = \mathbf{v} \oplus \delta$, where $||\delta||_0 \leq k$, if $\mathbf{v}$ and $\mathbf{v}'$ satisfy Eq. (8) and Eq. (9) in Theorem 2, we have

$$\mathbb{P}(f(h(\mathbf{v}' \oplus \epsilon)) = c^+) > \max_{\mathbf{v}_i^- \in \mathbf{V}^-} \mathbb{P}(f(h(\mathbf{v}' \oplus \epsilon)) = c_i^-), \ \forall \, \|\delta\|_0 \leq k, \tag{C.41}$$

which implies provable l_0^k-certified robustness retention of h at $(\mathbf{v}, c^+)$ in downstream tasks. The proof is completed.

D Training Algorithms

We summarize the training method of Sec. 5.1 for training smoothed GNN encoders in Algorithm 1. Specifically, at each training epoch, we first generate two augmented graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ via $q_i(\mathcal{G})$ and $q_j(\mathcal{G})$, respectively, where $q_i(\mathcal{G})$ and $q_j(\mathcal{G})$ are two graph augmentations sampled from an augmentation pool $\mathcal{T}$. The graph augmentation includes edge perturbation, feature masking, node dropping, etc. (line 3). Then we inject randomized edgedrop noise ϵ to one of the augmented graphs $\mathcal{G}_i$ (line 4). From line 5 to line 10, we train the GNN encoder h_θ through GCL by maximizing the agreement of representations in these two views. In detail, we apply h_θ to obtain node or graph representations $\mathbf{Z}_i$ and $\mathbf{Z}_j$ from $\mathcal{G}_i$ and $\mathcal{G}_j$, respectively (line 5), then we do gradient descent on θ based on Eq. (1).

E Discussions

E.1 Difference between RES and Edge-dropping Augmentations in [9] and [58]

Our RES is inherently different from the random edge-dropping augmentation in GraphCL [9] and the learnable edge-dropping augmentation in ADGCL [58]: (**i**) Random edge-dropping is an augmentation method to generate different augmented views and maximize the agreement between views, and the learnable edge-dropping [58] is also an augmentation method to enhance downstream task performance. However, RES is devised from the robustness perspective, providing certifiable robustness and enhancing the robustness of any GCL method. (**ii**) While random edge-dropping and learnable edge-dropping are only applied to augment graphs for GCL, RES extends beyond this. Following the generation of two augmented views as shown in Sec. 5.1, RES injects randomized edgedrop noise into one augmented view during GCL training. Then, it performs randomized edgedrop smoothing in the inference phase through Monte Carlo, as shown in Sec. 5.2. Specifically, for inference using RES, μ samples of $h(\mathbf{v} \oplus \epsilon)$ are drawn by injecting randomized edge-drop noise ϵ to $\mathbf{v}$ μ times. The final prediction is from Monte Carlo, selecting the μ predictions with the highest frequency in μ samples.

To demonstrate the effectiveness of RES, we further compare ADGCL with RES-GraphCL on MUTAG and PROTEINS. We also add RES-ADGCL into comparisons. More details are shown in Appendix. I.4.

E.2 Additional Details of Concatenation Vector

The concatenation vector $\mathbf{v}$ is a vector to depict the structure of the node/graph for learning representations. For node-level tasks, it represents the connection status of any pair of nodes in the K-hop

subgraph of the node v. For graph-level tasks, it represents the connection status of any pair of nodes in the graph $\mathcal{G}$. To construct such a vector, we select the upper triangular part of the adjacency matrix of the K-hop subgraph of v or the graph $\mathcal{G}$ and flatten it into the vector, where each item in this vector can denote the connection status of any pair of nodes in the K-hop subgraph of the node v or the graph $\mathcal{G}$.

The motivation for using this notation is that since we focus on perturbations on the graph structure $\mathbf{A}$ in this paper, we treat the feature vector of v as a constant and use the adjacency matrix of the K-hop subgraph of the node or the adjacency matrix of the graph to represent the structure of the node or graph. For simplicity and clarity, given a GNN encoder h and the concatenation vector $\mathbf{v}$ of the node v or the graph $\mathcal{G}$ as above, we then omit the node feature matrix $\mathbf{X}$ and simply write the node v's representation $h_v(A, X)$ and the graph $\mathcal{G}$'s representation $h(\mathcal{G})$ as $h(\mathbf{v})$. Therefore, we use a unified notation $\mathbf{v}$ to denote the node v or the graph $\mathcal{G}$, and further facilitate our theoretical derivations.

E.3 Definition of Well-trained GNN encoders

The well-trained GNN encoder is defined as an encoder that can extract meaningful and discriminative representations by mapping the positive pairs closer in the latent space while pushing dissimilar samples away.

To evaluate whether a GNN encoder h is well trained or not mathematically, we introduce criteria based on the similarity between node/graph representations in the latent space. For each positive pair $(\mathbf{v}, \mathbf{v}^+)$ with its negative samples $\mathbf{V}^- = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$, we clarify that h is well-trained at $(\mathbf{v}, \mathbf{v}^+)$ if the following inequality is satisfied:

$$s(h(\mathbf{v}), h(\mathbf{v}^+)) > \max_{\mathbf{v}^- \in \mathbf{V}^-} s(h(\mathbf{v}), h(\mathbf{v}^-)), \tag{E.42}$$

where $s(\cdot, \cdot)$ is a cosine similarity function. This implies that h can effectively discriminate $\mathbf{v}$ from all its negative samples in $\mathbf{V}^-$ and learn the meaningful representations for $\mathbf{v}$ in the latent space. Therefore, based on Eq. (E.42), we can further extend the criteria for certifying robustness in GCL, which is shown in Definition 2.

E.4 Rationale behind Setting High Values for β

The proposed robust encoder training method in Sec. 5.1 improves the model utility of GCL. Even setting a large β for RES, we can still obtain high robust accuracy on clean graphs, further leading to high certified accuracies.

Specifically, as shown in Sec. 6.1, certified accuracy denotes the fraction of correctly predicted test nodes/graphs whose certified perturbation size is not smaller than the given perturbation size. It implies that these certified robust samples should also be correctly predicted by RES in the clean datasets. However, as the reviewer said, introducing randomized edgedrop solely to test samples during the inference could potentially hurt downstream task performance and further negatively impact the certified robustness based on Eq.(8). Thus, we propose robust encoder training for RES in Sec. 5.1 by injecting randomized edgedrop noise into one augmented view during GCL . It ensures the samples with randomized edgedrop noises align in latent class with clean samples under the encoder, thereby mitigating the negative impacts of such noises and further benefiting the robustness and certification of RES.

F Code

Our code is available at `https://github.com/ventr1c/RES-GCL`.

G Additonal Details of Experiment Settings

G.1 Dataset Statistics

For node classification, we conduct experiments on 4 public benchmark datasets: Cora, Pubmed [47], Amazon-Computers [48] and OGB-arxiv [49], Cora and Pubmed are small citation networks. Amazon-Computers is a network of goods represented as nodes, and edges between nodes represent

that the two goods are frequently bought together. OGB-arxiv is a large-scale citation network. For graph classification, we use 3 well-known dataset: MUTAG, PROTEINS [50] and OGB-molhiv [49]. MUTAG is a collection of nitroaromatic compounds. PROTEINS is a set of proteins that are classified as enzymes or non-enzymes. OGB-molhiv is a dataset contains molecules, which is adopted from MoleculeNet [59]. Among these datasets, for Cora and Pubmed, we evaluate the models on the public splits. Regarding to the other 5 datasets, Coauthor-Physics, OGB-arxiv, MUTAG, PROTEINS and OGB-molhiv, we instead randomly select 10%, 10%, and 80% nodes/graphs for the training, validation and test, respectively. The statistics details of these datasets are summarized in Table 2.

Table 2: Dataset Statistics

Datasets	#Graphs	#Avg. Nodes	#Avg. Edges	#Avg. Feature	#Classes
Cora	1	2,708	5,429	1,443	7
Pubmed	1	19,717	44,338	500	3
Coauthor-Physics	1	34,493	495,924	8415	5
OGB-arxiv	1	169,343	1,166,243	128	40
MUTAG	188	17.9	39.6	7	2
PROTEINS	1,113	39.1	145.6	3	2
OGB-molhiv	41,127	25.5	27.5	9	2

G.2 Attack Methods

One of our goals is to show RES is robust to various structural noises, we evaluate RES on 4 types of structural attacks in evasion setting, i.e., Random attack, Nettack [11], PRBCD [51], CLGA [16] for both node and graph classification. The procedure of the evasion attack against GCL in transductive node classification is shown in Algorithm 2. The details of these attacks are described following:

1. **Random Attack**: We randomly add some noisy edges to the graphs for node classification and graph classification, respectively, until the attack budget is satisfied. Specifically, we consider two kinds of attack settings for node classification, that is, global random attack and targeted random attack. For global random attack, which is used in Sec. 6.2, we randomly inject some fake edges (10% in our setting) into the whole graph. For targeted attack, we randomly connect some fake edges to the direct neighbors of target nodes.
2. **Nettack** [11]: It is a targeted attacks for node classification that manipulate the graph structure to mislead the prediction of target nodes.
3. **PRBCD** [51]: It is a scalable global attack for node classification that aims to decrease the overall accuracy of the graph.
4. **CLGA** [16]: It is an unsupervised gradient-based poisoning attack targeting graph contrastive learning for node classification. Since we focus on evasion attacks, we directly use the poisoned graph generated by CLGA in the downstream tasks to evaluate the performance and regard it as an evasion global attack.

Moreover, we further consider two graph injection attack methods, i.e., TDGIA [60] and AGIA [61], as baselines to demonstrate the robustness of RES. The details of experimental results are shown in Sec. I.3.

G.3 Compared Methods

We select four state-of-the-art GCL methods and employ RES on them to train smoothed GNN encoders:

1. **GRACE** [30]: It is node-level GCL method which creates multiple augmented views by perturbing graph structure and masking node features. Then it encourages consistency between the same nodes in different views.
2. **BGRL** [52]:Insipred by BYOL [62], it is performs graph contrastive learning that does not require negative samples. Specifically, it applies two graph encoders (i.e., online and targeted encoders), and update them iteratively to make the predicted representations closer to the true representations for each node.

Algorithm 2 The Evasion Attack Procedure against GCL in Transductive Node Classification

Input: Clean graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, GNN encoder h_θ, target node v with class label c, perturbation δ, downstream classifier f.
Output: success (i.e., attack v successfully or not)
1: Train h_θ on $\mathcal{G}$ via GCL;
2: Obtain perturbed node v' by adding perturbation δ to v;
3: Generate the representation of v' as $h(v')$;
4: **if** $f(h(v')) = c$ **then**
5: **return** success $\leftarrow$ **false**;
6: **else**
7: **return** success $\leftarrow$ **true**;
8: **end if**

3. **DGI** [31]: It is a state-of-the-art GCL method which adapted from Deep InfoMax [63] to maximize the mutual information between local and global features.
4. **GraphCL** [9]: It is the first work to study GCL at graph-level. Specifically, it constructs four types of graph augmentations and adapts SimCLR [55] to learn graph-level embeddings.

Moreover, we also compare RES-GCL with several representative and state-of-the-art graph representation learning methods and robust GCLs against structural noises:

1. **Node2Vec** [53]: It is a traditional unsupervised methods. Its key idea is to perform random walks on the graph to generate sequences of nodes that capture both the local and global structure of the graph.
2. **GAE** [54]: It is a representative unsupervised learning method which learns a low-dimensional representation of a graph by encoding its nodes and edges into a vector space, and then decoding this representation back into a graph structure. It is trained to minimize the reconstruction error between the original graph and the reconstructed graph.
3. **GCL-Jaccard**: It is implemented by removing dissimilar edges based on Jaccard similarity before and after the training phase, respectively, which is inspired from GCN-Jaccard [64].
4. **Ariel** [15]: It is a robust GCL method which uses an additional adversarial graph view in graph contrastive learning to improve the robustness. An information regularizer is also applied to stabilize its performance.

G.4 Implementation details

A 2-layer GCN is employed as the backbone GNN encoder and a common used linear evaluation scheme [31] is adopt in the downstream tasks. More specifically, each GNN encoder is firstly trained via GCL method and then the resulting embeddings are used to train and test a l_2-regularized logistic regression classifier. GRACE is implemented based on the source code published by authors [1]. BGRL and DGI methods are implemented based on PyGCL library [65]. All hyperparameters of all methods are tuned based on the validation set for fair comparison. All models are trained on an A6000 GPU with 48G memory.

G.5 Attack Settings

In this paper, we assume that the attacker can conduct attack in two settings: transductive node classification and inductive graph classification. These two settings are described below:

- **Transductive Setting:** In this setting, test instances (nodes/graphs) are visible during both training the GNN encoder and inference in downstream tasks. Specifically, we first train a GNN encoder h via GCL on a clean dataset that includes test nodes to generate node representations. Then, an attacker adds perturbations to the test nodes, causing h to produce poor representations that degrade performance on downstream tasks. For example, an attacker may attempt to manipulate a social network by injecting fake edges, which could affect the performance of

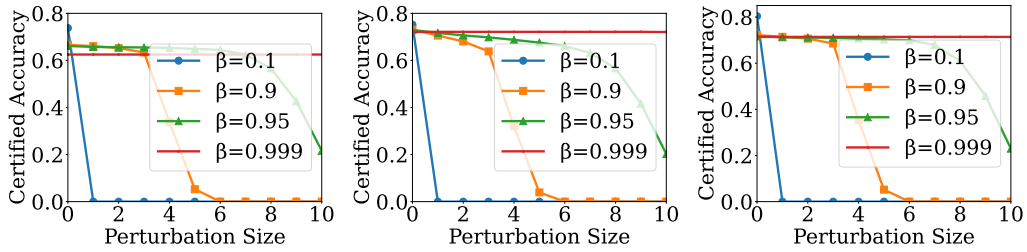
[1] https://github.com/CRIPAC-DIG/GRACE

a well-trained h in tasks such as community detection, influential node identification, or link prediction.

- **Inductive Setting:** In this setting, test instances only appear in downstream tasks and are invisible during the training phase. This setting is similar to the transductive setting, but the test nodes/graphs are not seen during training. This scenario commonly arises in real-world applications such as new drug discovery, where an attacker may attempt to manipulate a new molecular graph in the test set to mislead the model, resulting in incorrect predictions in downstream tasks.

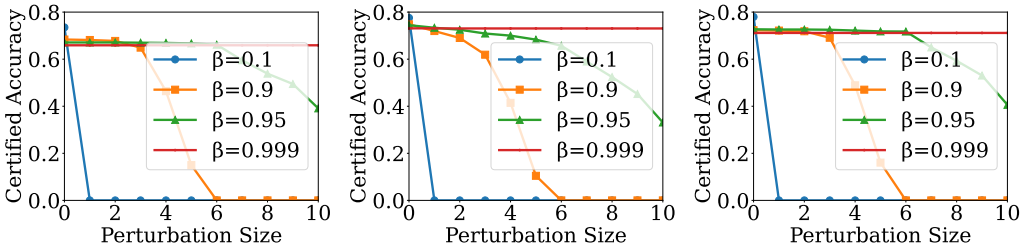
H Additional Results of the Performance of Certificates

In this section, we extend the experiments in Sec. 6.3 and present comprehensive results on the performance of robustness certificates. Our aim is to demonstrate that RES can effectively provide certifiable robustness for various GCL methods. We select GRACE, BGRL, DGI, and GraphCL as the target GCL methods and integrate them with RES. We perform experiments on Cora, Pubmed, Coauthor-Physics, and OGB-arxiv for node classification tasks, as well as MUTAG, PROTEINS, and OGB-molhiv for graph classification tasks. *Certified accuracy* is selected as the evaluation metric. Specifically, *certified accuracy* [46, 24] denotes the fraction of correctly predicted test nodes/graphs whose certified perturbation size is no smaller than the given perturbation size. The complete results are reported in Fig. 4 to Fig. 8.



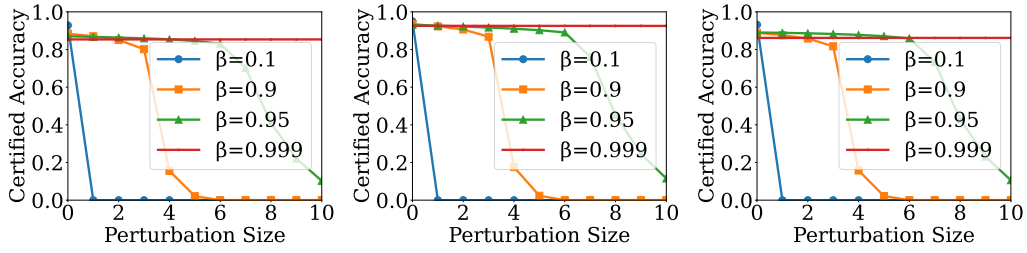
(a) GRACE (b) BGRL (c) DGI

Figure 4: Certified accuracy of RES-GCL on Cora



(a) GRACE (b) BGRL (c) DGI

Figure 5: Certified accuracy of RES-GCL on Pubmed

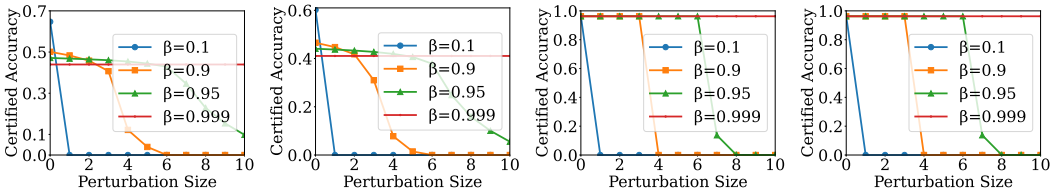


(a) GRACE (b) BGRL (c) DGI

Figure 6: Certified accuracy of RES-GCL on Coauthor-Physics

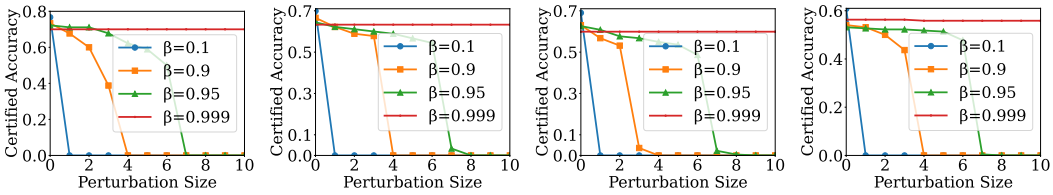
I Additional Results of the Performance of Robustness

In this section, we provide additional experimental results to further showcase the effectiveness of the RES in enhancing the robustness of GCL against various adversarial attacks. More specifically,



(a) GRACE, OGB-arxiv (b) DGI, OGB-arxiv (c) GraphCL, OGB-molhiv (d) BGRL, OGB-molhiv,

Figure 7: Certified accuracy of smoothed GCL on OGB-arxiv and OGB-molhiv



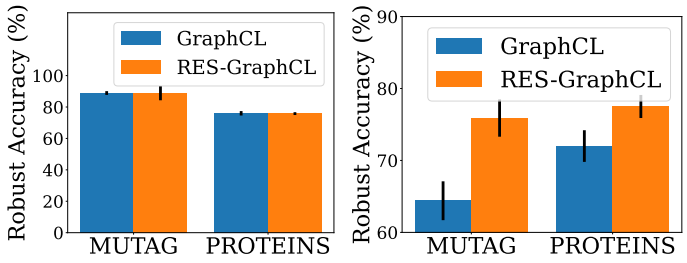
(a) GraphCL, MUTAG (b) BGRL, MUTAG (c) GraphCL, PROTEINS (d) BGRL, PROTEINS

Figure 8: Certified accuracy of smoothed GCL on MUTAG and PROTEINS

in Sec. I.1, we show the comparison results of RES with the baselines on graph classification. In Sec. I.2, we conduct experiments to demonstrate GCL with RES is resistant to different levels of structural noises. In Sec. I.3, we present the comparison results of three target GCL methods (i.e., GRACE, BGRL, DGI) against three types of structural attacks (i.e., Random, CLGA and PRBCD) on node classification. In Sec. I.4, we present additional experimental results of two advanced GCL methods (i.e., ADGCL [58] and RGCL [66]) on graph classification.

I.1 Robust Performance on Graph Classification

In this subsection, we conduct experiments on graph classification to demonstrate the effectiveness of our method in this downstream task. Due to the limited availability of open-source attack methods specifically designed for graph classification, we utilize random attacks as the attack method. However, we believe that our method is robust against other attack methods as well. Specifically, we select GraphCL as the target GCL methods. The perturbation rate of random attack is 0.1. The smoothed version of GraphCL is denoted as RES-GraphCL. The results on MUTAG and PROTEINS are given in Fig. 9. From the figure, we observe: **(i)** When no attack is applied to the raw graphs, RES-GraphCL achieves comparable performance to the baseline GraphCL. **(ii)** When attacks are conducted on the noisy graphs, RES-GraphCL consistently outperforms the baseline on both the MUTAG and PROTEINS datasets. This result demonstrates the effectiveness of our method in enhancing the robustness of GraphCL against adversarial attacks.

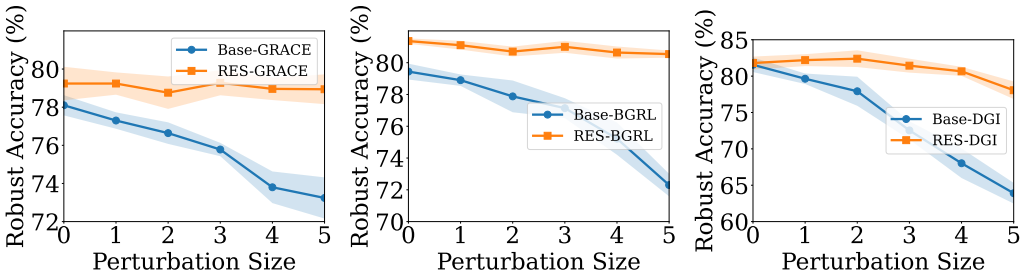


(a) Clean (b) Noisy

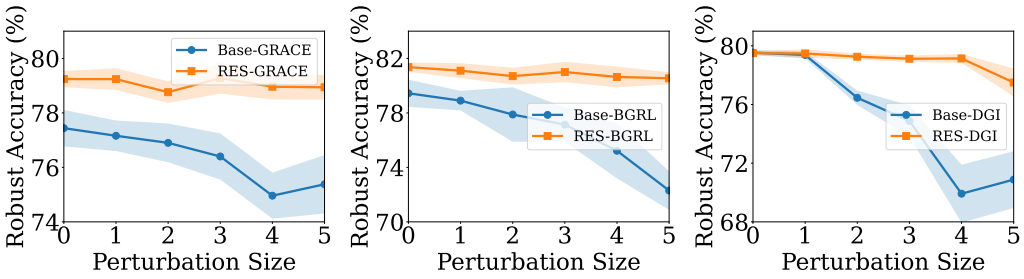
Figure 9: Robust accuracy of GraphCL for graph classification against random attack

I.2 Robustness Under Different Noisy Levels.

To demonstrate the ability of our method to improve the robustness of Graph Contrastive Learning (GCL) against different levels of structural noise, we compare the robust accuracy of the GCL methods w/o applying our RES under evasion attacks for node classification. Specifically, we select GRACE, BGRL and DGI as target GCL methods. We set $(1-\beta)=0.05$ and $\mu=50$. We consider two targeted attack method, random attack and Nettack [11] to conduct targeted attacks. The attack budget is set from 0 to 5. We randomly select 15% of the test nodes as the target nodes to compute the robust accuracy. The results on the Cora and Pubmed datasets are reported in Fig. 10 and Fig. 11. We observe: **(i)** The robust accuracies of the baseline methods exhibit a significant drop as the

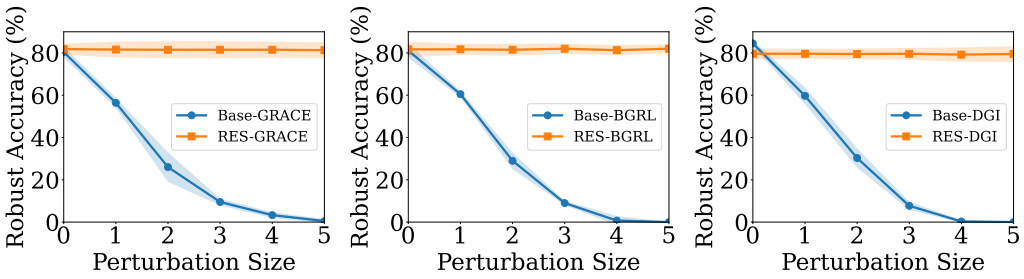


(a) Cora, GRACE (b) Cora, BGRL (c) Cora, DGI

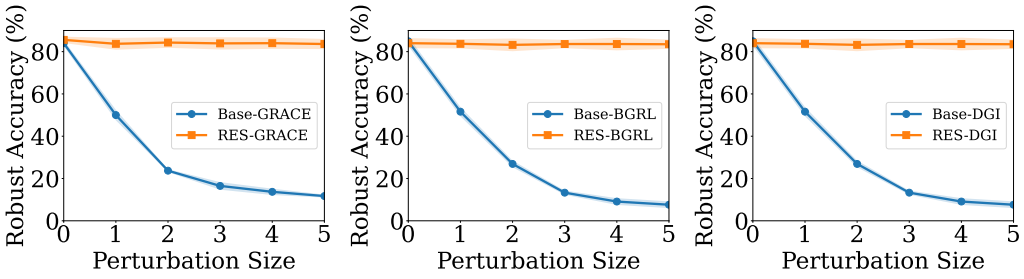


(d) Pubmed, GRACE (e) Pubmed, BGRL (f) Pubmed, DGI

Figure 10: Robust accuracy under different perturbation sizes of random attack



(a) Cora, GRACE (b) Cora, BGRL (c) Cora, DGI



(d) Pubmed, GRACE (e) Pubmed, BGRL (f) Pubmed, DGI

Figure 11: Robust accuracy under different perturbation sizes of Nettack

perturbation sizes increase, which is expected. In contrast, the performances of RES are much more stable and consistently outperform the baselines. This demonstrates the robustness of GCL with RES against various levels of structural noise. **(ii)** The high robust accuracies of the three GCL models demonstrate that our RES is effectively applicable to various GCL method.

I.3 Additonal Results of Robust Performance on Node Classification

In this subsection, we provide additional experimental results from Section 6.2, focusing on node classification. We select GRACE, BGRL, and DGI as the target GCL methods and evaluate their performance on four types of graphs: raw graphs, random attack perturbed graphs, CLGA perturbed graphs, and PRBCD perturbed graphs. The perturbation ratio is set to 0.1. Table 3 presents the results of these experiments. The highlighted results denote the best performance for each pair of GCL and RES-GCL. Note that BGRL, implemented based on PyGCL [65], encounters out-of-memory (OOM) errors in our platform, and hence the results for BGRL on OGB-arxiv are left blank. From

the table, we observe that all three GCL methods, when combined with RES, achieve state-of-the-art performance. This demonstrates the effectiveness of our method in enhancing the robustness of various GCL methods. By incorporating RES, the GCL models are more resilient to adversarial attacks and exhibit improved performance across different types of perturbed graphs.

Moreover, we further consider two graph injection attack methods, i.e., TDGIA [60] and AGIA [61], as baselines to demonstrate the robustness of RES. Specificially, we select GRACE as the target GCL methods and evaluate them on three types of graphs: raw graphs, TDGIA perturbed graphs and AGIA perturbed graphs. We insert the same number of fake nodes as the target nodes. We set $(1-\beta)=0.1$ and $\mu=50$. The comparison results on four datasets are shown in Table. 4. From the results, we observe that (**i**) RES-GRACE consistently outperforms the baselines across 4 datasets in defending graph injection attacks. (**ii**) Both TDGIA and AGIA are much more powerful than the structural attack methods in Table 3 against GCL.

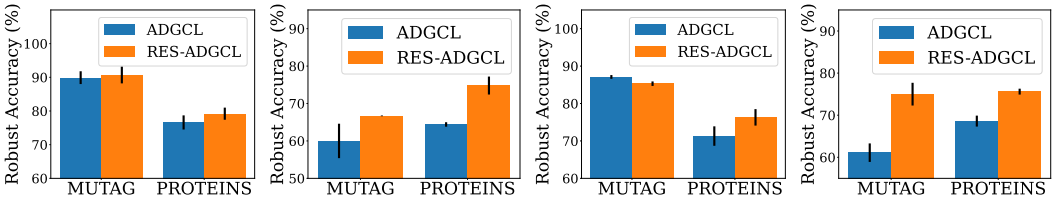
Table 3: Robust accuracy results of GCL methods for node classification.

Dataset	Graph	GRACE	RES-GRACE	BGRL	RES-BGRL	DGI	RES-DGI
Cora	Raw	77.1±1.6	**79.7±1.0**	78.5±1.6	**79.9±1.2**	81.3±0.7	**81.4±0.8**
	Random	74.5±2.1	**79.7±1.0**	76.2±1.2	**79.6±1.1**	77.5±1.0	**79.2±0.6**
	CLGA	74.9±2.0	**78.2±1.0**	75.8±1.6	**79.4±0.9**	79.5±0.5	**80.9±1.0**
	PRBCD	75.8±2.5	**78.5±1.7**	76.4±0.6	**79.5±0.8**	79.6±1.4	**80.9±1.1**
Pubmed	Raw	79.5±2.9	**79.5±1.2**	79.9±1.4	**81.5±0.6**	**80.1±0.9**	80.0±0.8
	Random	75.0±1.0	**78.2±0.9**	74.0±1.0	**81.0±0.7**	76.7±0.7	**78.8±0.6**
	CLGA	76.6±2.5	**78.3±1.1**	77.9±0.3	**81.6±0.2**	79.6±0.6	**80.0±1.2**
	PRBCD	73.2±2.3	**78.8±1.7**	71.6±2.4	**80.9±0.5**	75.4±0.9	**78.0±0.6**
Physics	Raw	94.0±0.4	**94.7±0.2**	95.3±0.1	**95.6±0.1**	93.5±0.6	**94.1±0.3**
	Random	92.6±0.5	**94.2±0.3**	94.0±0.2	**95.5±0.2**	91.5±0.9	**92.2±0.4**
	PRBCD	89.2±0.6	**94.1±0.2**	92.3±0.2	**95.4±0.2**	88.8±0.5	**90.0±0.4**
OGB-arxiv	Raw	65.1±0.5	**65.2±0.1**	-	-	**65.0±0.2**	64.8±0.1
	Random	59.0±0.2	**60.0±0.1**	-	-	58.0±0.1	**58.9±0.1**
	PRBCD	55.7±0.4	**58.3±0.4**	-	-	56.9±0.9	**57.2±0.3**

Table 4: Robust accuracy results of GCL methods against injection attacks.

Dataset	Graph	GRACE	RES-GRACE
Cora	Raw	77.1±1.6	**79.7±1.0**
	TDGIA	22.4±1.5	**78.7±1.1**
	AGIA	21.1±1.4	**78.4±0.9**
Pubmed	Raw	79.5±3.1	**79.5±1.2**
	TDGIA	50.6±1.2	**77.0±0.4**
	AGIA	50.2±1.6	**77.5±1.0**
Physics	Raw	94.0±0.4	**94.7±0.2**
	TDGIA	52.2±1.0	**93.8±0.1**
	AGIA	52.3±0.5	**94.0±0.2**
OGB-arxiv	Raw	65.1±0.5	**65.1±0.1**
	TDGIA	40.3±0.4	**46.9±0.1**
	AGIA	40.4±0.3	**47.0±0.1**

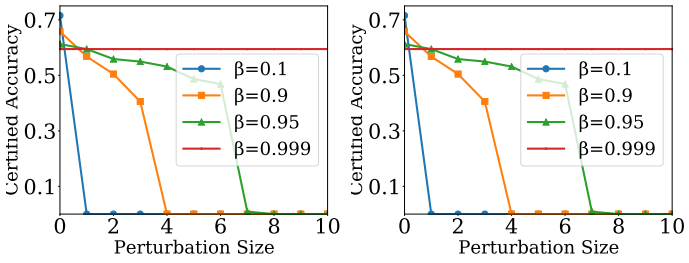
I.4 Additional Results of RES for Advanced GCLs



(a) ADGCL, Clean (b) ADGCL, Noisy (c) RGCL, Clean (d) RGCL, Noisy

Figure 12: Robust accuracy of ADGCL and RGCL for graph classification against random attack

To further demonstrate the effectiveness of RES, we further add ADGCL [58] and RGCL [66] as baselines and implement RES-ADGCL and RES-RGCL. We set graph classification as the downstream task. The hyperparameter is tuned based on the performance of the validation set. We use random attack to get the noisy graphs and the perturbation rate is 0.1. Each experiment is conducted 5 times and the average results are reported. Comparison results on MUTAG and PROTEINS are

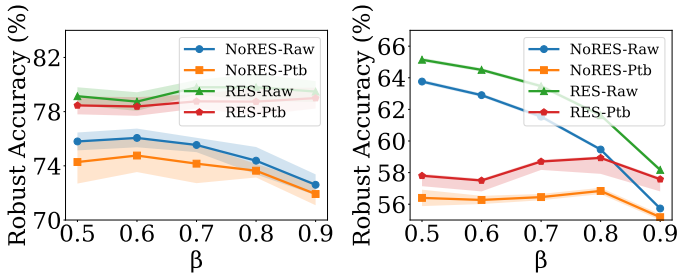


(a) MUTAG (b) PROTEINS

Figure 13: Certified accuracy of RGCL on MUTAG and PROTEINS

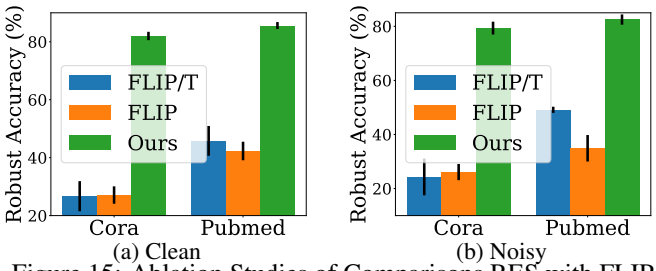
shown in Fig. 12. From this figure, we observe that all RES-GCL methods achieve comparable performances to the baselines on raw graphs and consistently outperform the baselines in the noisy graphs of two datasets, which validates the effectiveness of RES in any GCL model.

We also report the certified accuracy of RES-RGCL on the two datasets. The results are shown in Fig. 13. From the figure, we can observe there is a tradeoff between certified robustness and model utility, which is similar to that of Sec. 6.3.



(a) Cora (b) OGB-arxiv

Figure 14: Ablation Studies of Training with RES on Cora and OGB-arxiv



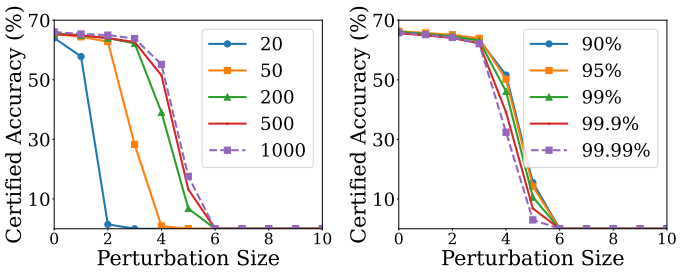
(a) Clean (b) Noisy

Figure 15: Ablation Studies of Comparisons RES with FLIP

J Additional Results of Ablation Studies

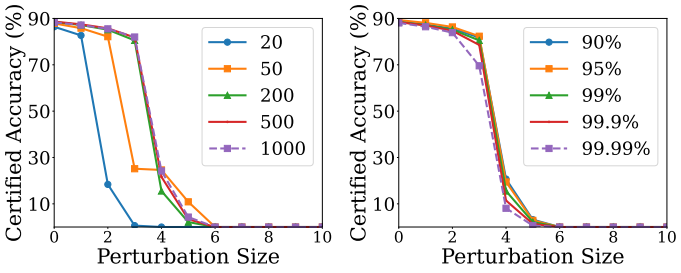
J.1 Ablation Studies on Training with RES

In this subsection, we conduct ablation studies to investigate the effect of our training method for robust GCL. To demonstrate that our training method improves the robustness of GCL, we do not inject random edgedrop noise during the training phase and obtain a variant called NoRES. We select PRBCD as the attack method to generate noisy graphs, and set $\mu = 50$. We vary the value of β as $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ and compare the robust accuracy of RES and NoRES on both raw graphs and PRBCD-perturbed graphs. The results of the variant NoRES on raw and PRBCD-perturbed graphs are denoted as NoRES-Raw and NoRES-Ptb, respectively. We report the results on the Cora and OGB-arxiv datasets in Fig. 14. From the figure, we observe the following: **(i)** RES consistently outperforms NoRES on both raw and perturbed graphs in all settings, implying the effectiveness of our training method for robust GCL. **(ii)** The variance of RES is lower than that of NoRES. This is because we inject randomized edgedrop noise into the graphs during the training phase, helping the models better understand and generalize from the data with randomized edgedrop noise, thereby ensuring the robustness and utility of the models.



(a) μ (b) $(1-\alpha)$

Figure 16: Hyperparameter Sensitivity Analysis on Cora



(a) μ (b) $(1-\alpha)$

Figure 17: Hyperparameter Sensitivity Analysis on Coauthor-Physics

J.2 Additional Results of Comparisons RES with FLIP

In this subsection, we present additional experimental results in Sec. 6.4 to further demonstrate that our RES method is significantly more effective in GCL compared to vanilla randomized smoothing [24]. We introduce two variants of our model, namely FLIP and FLIP/T, which replace the randomized edgedrop noise with binary random noise [24]. This noise flips the connection status within the graph with a probability of β. The key difference between FLIP and FLIP/T is that FLIP injects binary random noise into the graphs during both the training and inference phases, while FLIP/T only injects binary random noise during the inference phase. For our RES method, we set $\beta = 0.9$, and $\mu = 50$. For FLIP and FLIP/T, to ensure a fair comparison, we set $\mu = 50$, and vary β over $\{0.1, 0.2, \cdots, 0.9\}$ and select the value that yields the best performance on the validation set of clean graphs. We select GRACE as the targeted GCL method and compare the robust accuracy of our RES with FLIP and FLIP/T on the clean and noisy graph under Nettack with an attack budget of 3. The average robust accuracy and standard deviation on Cora and Pubmed datasets are reported in Fig. 15. We observe: **(i)** RES consistently outperforms FLIP and FLIP/T on clean and noisy graphs of Cora and Pubmed datasets, further validating the effectiveness of RES in providing certified robustness for GCL. **(ii)** FLIP and FLIP/T exhibit comparable performance on all graphs, but significantly lower than RES. This finding confirms our analysis that vanilla randomized smoothing introduces numerous spurious/noisy edges to the graph, resulting in poor representation learning by the GNN encoder and compromising downstream task performance.

K Hyperparameter Sensitivity Analysis

We further investigate how hyperparameter $(1-\alpha)$ and μ affect the performance of robustness certificates of our RES, where $(1-\alpha)$ and μ control the confidence level and the number of Monte Carlo samples used to compute the certified accuracy. We vary the value of $(1-\alpha)$ as $\{90\%, 95\%, 99\%, 99.9\%, 99.99\%\}$ and fix μ as 200, vary μ as $\{20, 50, 200, 500, 1000\}$ and fix $(1-\alpha)$ as 99%, respectively. β is set as 0.9. We report the certified accuracy of RES-GRACE on Cora and Coauthor-Physics dataset in Fig. 16 and Fig. 17. From the figures, we observe: **(i)** As μ increases, the certified accuracy curve becomes higher. The reason for this is that a larger value of μ makes the estimated probability bound $\underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon)$ and $\overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon)$ in Eq. (8) tighter, resulting in a higher certified perturbation size of a test sample. **(ii)** As $(1-\alpha)$ increases, the certified accuracy curve becomes slightly lower. This is because a higher confidence level leads to a looser estimation of $\underline{p_{\mathbf{v}^+,h}}(\mathbf{v} \oplus \epsilon)$ and $\overline{p_{\mathbf{v}_i^-,h}}(\mathbf{v} \oplus \epsilon)$ in Eq. (8), meaning that fewer nodes satisfy Theorem 2 under the same perturbation size, resulting in a smaller certified perturbation size of a test sample.

L Limitations

In this paper, we propose a unified criteria to evaluate and certify the robustness of GCL. Our proposed approach, Randomized Edgedrop Smoothing (RES), injects randomized edgedrop noise into graphs to provide certified robustness for GCL on unlabeled data. Moreover, we design an effective training method for robust GCL by incorporating randomized edgedrop noise during the training phase. The theoretical analysis and extensive experiments show the effectiveness of our proposed RES.

Limitation & future work: Our current results are limited mainly to GCL while we believe it is also interesting to develop new techniques to other graph self-supervised methods e.g. generative and neighborhood prediction methods based on our framework, which we leave for immediate future work. We hope that this work could inspire future certifiably defense algorithms of adversarial attacks. Additionally, in this paper, we only focus on the graph-structured data. Thus, it is also interesting to investigate how to extend it to other domains, e.g., images and texts. Furthermore, in this paper, we utilize Monte Carlo algorithms to calculate robustness certificates for GCL, potentially increasing the computational demands. Therefore, it is also worthwhile to investigate methods to improve the efficiency of the robustness certification for GCL. Due to the nature of this work, there may not be any potential negative social impact that is easily predictable.