REVISITING ROBUSTNESS IN GRAPH MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many works show that node-level predictions of Graph Neural Networks (GNNs) are unrobust to small, often termed adversarial, changes to the graph structure. However, because manual inspection of a graph is difficult, it is unclear if the studied perturbations always preserve a core assumption of adversarial examples: that of unchanged semantic content. To address this problem, we introduce a more principled notion of an adversarial graph, which is aware of semantic content change. Using Contextual Stochastic Block Models (CSBMs) and real-world graphs, our results suggest: i) for a majority of nodes the prevalent perturbation models include a large fraction of perturbed graphs violating the unchanged semantics assumption; ii) surprisingly, all assessed GNNs show *over-robustness* - that is robustness *beyond* the point of semantic change. We find this to be a complementary phenomenon to adversarial examples and show that including the label-structure of the training graph into the inference process of GNNs significantly reduces over-robustness, while having a positive effect on test accuracy and adversarial robustness. Theoretically, leveraging our new semantics-aware notion of robustness, we prove that there is no robustness-accuracy tradeoff for inductively classifying a newly added node.

1 INTRODUCTION

Graph Neural Networks (GNNs) are seen as state of the art for various graph learning tasks (Hu et al., 2020; 2021). However, there is strong evidence that GNNs are unrobust to changes to the underlying graph (Zügner et al., 2018; Geisler et al., 2021). This has led to the general belief that GNNs can be easily fooled by adversarial examples and many works trying to increase the robustness of GNNs through various defenses (Günnemann, 2022). Originating from the study of deep image classifiers (Szegedy et al., 2014), an adversarial example has been defined as a small perturbation, usually measured using an ℓ_p-norm, which does not change the semantic content (i.e. category) of an image, but results in a different prediction. These perturbations are often termed unnoticeable relating to a human observer for whom a normal and an adversarially perturbed image are nearly indistinguishable (Goodfellow et al., 2015; Papernot et al., 2016). However, compared to visual tasks, it is difficult to visually inspect (large-scale) graphs. This has led to a fundamental question:

What constitutes a small, semantics-preserving perturbation to a graph?

The de facto standard in the literature is to measure small changes to the graph's structureusing the ℓ_0-pseudonorm (Zheng et al., 2021; Günnemann, 2022). Then, the associated threat models restrict the total number of inserted and deleted edges globally in the graph and/or locally per node. However, if the observation of semantic content preservation for these kind of perturbation models transfers to the graph domain can be questioned: Due to the majority of low-degree nodes in real-world graphs, small ℓ_0-norm restrictions still allow to completely remove a significant number of nodes from their original neighbourhood. Only few works introduce measures beyond ℓ_0-norm restrictions. In particular, it was proposed to additionally use different global graph properties as a proxy for unnoticability, such as the degree distribution (Zügner et al., 2018), degree assortativity (Li et al., 2021), or other homophily metrics (Chen et al., 2022). While these are important first steps, the exact relation between preserving certain graph properties and the graph's semantic content (e.g. node-categories) is unclear. For instance, one can completely rewire the graph by iter-

atively swapping the endpoints of two randomly selected edges and preserve the global degree distribution. As a result, current literature lacks a principled understanding of semantics-preservation in their employed notions of smallness as well as robustness studies using threat models only including provable semantics-preserving perturbations to a graph.

We bridge this gap by being the first to directly address the problem of exactly measuring (node-level) semantic content preservation in a graph under structure perturbations. Surprisingly, using Contextual Stochastic Block Models (CSBMs), this leads us to discover a novel phenomenon: GNNs show strong robustness *beyond* the point of semantic change (see Figure 1). This does *not* contradict the existence of adversarial examples for the same GNNs. Related to the small degree of nodes, we find that common perturbation sets include both: graphs which are truly adversarial as well as graphs with changed semantic content. Our contributions are:

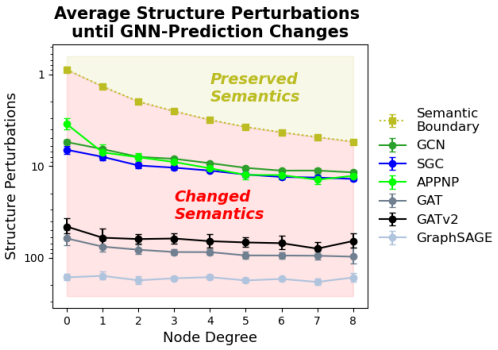


Figure 1: Data from CSBM graphs. The semantic boundary indicates when the semantic-content (i.e., the most likely class) of a node of particular degree changes on average. All GNNs show robustness *beyond* the point of semantic change.

1. We define a semantics-aware notion of adversarial robustness (Section 3) for node-level predictions. Using this, we introduce a novel concept: *over-robustness* - that is (unwanted) robustness against admissible perturbations with changed semantic content (i.e. changed ground-truth labels).

2. Using CSBMs, we find: i) common perturbations sets, next to truly adversarial examples, include a large fraction of graphs with changed semantic content (Section 5.1); ii) all examined GNNs show significant *over-robustness* to these graphs (Section 5.2) and we observe similar patterns on real-world datasets (Section 5.2.1). Using ℓ_0-norm bounded adversaries on CSBM graphs, we find a considerable amount of a conventional adversarial robustness to be in fact over-robustness.

3. Including the known label-structure through Label Propagation (LP) (Huang et al., 2021) into the inference process of GNNs significantly reduces over-robustness with no negative effects on test accuracy or adversarial robustness (Section 5.2) and similar behaviour on real-world graphs.

4. Using semantic awareness, we prove the existence of a model achieving both, optimal robustness and accuracy in classifying an inductively sampled node (Section 4.1), i.e. no robustness-accuracy tradeoff for a non-i.i.d. data setting.

2 PRELIMINARIES

Let n be the number of nodes and d the feature dimension. We denote the node feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the (symmetric) adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, and the node labels $y \in \{0,1\}^n$ of which $y_L \in \{0,1\}^l$, $l \leq n$ are known. We assume a given graph has been sampled from a graph data generating distribution $\mathcal{D}_n$ denoted as $(\mathbf{X}, \mathbf{A}, y) \sim \mathcal{D}_n$. If it aids presentation, we abbreviate the graph without labels as $\mathcal{G} = (\mathbf{X}, \mathbf{A})$. We study inductive node classification (Zheng et al., 2021). Due to the non-i.i.d data generation, a node-classifier f may depend its decision on the whole known graph $(\mathbf{X}, \mathbf{A}, y_L)$. As a result we write $f(\mathbf{X}, \mathbf{A}, y_L)_v$ to denote the classification of a node v.

Graph Neural Networks. All GNNs covered in this work are a function $f(\mathbf{X}, \mathbf{A})$ of the node features $\mathbf{X}$ and adjacency matrix $\mathbf{A}$. GNNs often follow a message passing scheme: The representation of node v in the l-th layer is given by $\mathbf{h}_v^{(l} = \sigma^{(l)}(\text{AGG}^{(l)}[\mathbf{h}_u^{(l-1)}\mathbf{W}^{(l)}, \forall\, u \in \mathcal{N}(v) \cup \{v\}])$ with neighbors $\mathcal{N}(v)$, an aggregation function AGG, non-linearity σ, and weights $[\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}]$.

Label Propagation. We use label spreading (Zhou et al., 2003), which builts a classifier $f(\mathbf{A}, y_L)$ by taking the row-wise $\arg\max$ of the iterate $F^t = \alpha\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}F^{t-1} + (1-\alpha)Y$, with $\mathbf{D}$ being the diagonal degree matrix; $Y \in \mathbb{R}^{n\times c}$ with $Y_{ij} = 1$ if $i \leq l$ and $y_L^i = j$, otherwise $Y_{ij} = 0$; and $\alpha \in [0,1]$. Similar to Huang et al. (2021), we combine LP and GNNs by replacing the zero-rows for $i > l$ in Y with GNN soft-predictions, effectively forming a function $f(\mathbf{X}, \mathbf{A}, y_L)$.

Adversarial Robustness. We study (inductive) evasion attacks, that is an adversary $\mathcal{A}$ can produce a perturbed graph $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in \mathcal{B}(\mathbf{X}, \mathbf{A})$ given the clean data $(\mathbf{X}, \mathbf{A})$ with the goal to fool a given node-classifier $f(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_L)_v \neq f(\mathbf{X}, \mathbf{A}, y_L)_v$ on nodes v. The perturbation set $\mathcal{B}(\mathbf{X}, \mathbf{A})$ collects all admissible perturbed graphs, defined by the threat model. We focus on direct structure attacks on a target node v (Zügner et al., 2018), setting $\mathcal{B}(\mathbf{X}, \mathbf{A}) = \{(\mathbf{X}, \tilde{\mathbf{A}}) \big| \|\mathbf{A}_{v,:} - \tilde{\mathbf{A}}_{v,:}\|_0 \leq \Delta\}$, with $\mathbf{A}_{v,:}$ referring to the v-th row of $\mathbf{A}$. This means the attacker can remove or add at most Δ edges to v.

Data Model. We leverage *Contextual Stochastic Block Models* (CSBMs) (Deshpande et al., 2018) to generate synthetic graphs with analytically tractable distributions. It defines edge-probabilities p between same-class nodes and q between different-class nodes and node-features are drawn from a Gaussian mixture model. Sampling from a CSBM can be understood as an iterative process over nodes $i \in [n]$: 1) Sample label $y_i \sim \text{Ber}(1/2)$ (Ber denoting the Bernoulli distribution). 2) Sample feature vector $\mathbf{X}_{i,:}|y_i \sim \mathcal{N}((2y_i - 1)\mu, \sigma\mathbf{I})$ with $\mu \in \mathbb{R}^d$, $\sigma \in \mathbb{R}$. 3) For all $j \in [n], j < i$ sample $\mathbf{A}_{j,i} \sim \text{Ber}(p)$ if $y_i = y_j$ and $\mathbf{A}_{j,i} \sim \text{Ber}(q)$ otherwise, and set $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$. We denote this process $(\mathbf{X}, \mathbf{A}, y) \sim \text{CSBM}_{n,p,q}^{\mu,\sigma^2}$. Inductively adding m nodes can be performed by repeating the above process for $i = n+1, \ldots, m$. Fountoulakis et al. (2022) show that depending on the distance of the means $d(-\mu, \mu)$, one can separate an easy regime, where a linear classifier ignoring $\mathbf{A}$ can perfectly separate the data and a hard regime, defined by $d(-\mu, \mu) = K\sigma$, with $0 < K \leq \mathcal{O}(\sqrt{\log n})$, where this is not possible. CSBMs are commonly used to study transductive tasks. Understanding the sampling process as an iteration allows to extend their application to inductive node-classification.

3 Revisiting Adversarial Perturbations

Given a clean graph $(\mathbf{X}, \mathbf{A}, y_L)$ and target node v. The perturbation set $\mathcal{B}(\mathbf{X}, \mathbf{A})$ comprises all possible (perturbed) graphs $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})$, which can be chosen by an adversary $\mathcal{A}$, with the goal to change the prediction of a node classifier f, i.e., $f(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_L)_v \neq f(\mathbf{X}, \mathbf{A}, y_L)_v$. The prevalent works implicitly assume that every $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in \mathcal{B}(\mathbf{X}, \mathbf{A})$ preserves the node-level semantic content of the clean graph, i.e. the original ground-truth label of v. If we would have an oracle Ω, which tells us the semantic content the known graph encodes about v, this assumption can be made explicit by writing $\Omega(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_L)_v = \Omega(\mathbf{X}, \mathbf{A}, y_L)_v$. Usually, we do not have access to such an oracle. However, we can try to model its behaviour by introducing a reference or base node classifier g. Then, the idea is to use g to indicate semantic content change and thereby, define the semantic boundary (see Figure 1). Exemplary, g could be derived from knowledge about the data generating process. We do so in Section 5, where we use the (Bayes) optimal classifier for CSBMs as g (see also Section 4). Note that labels themselves are often generated following a base classifier. Exemplary, this can be humans labelling selected nodes in a graph to generate a dataset for (semi-) supervised learning. Using a reference classifier g as a proxy for semantic content enables us to make a refined definition of an adversarial graph, which makes the unchanged-semantics assumption explicit:

Definition 1. *Let f be a node classifier and g a reference node classifier. Then the perturbed graph $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in \mathcal{B}(\mathbf{X}, \mathbf{A})$ chosen by an adversary $\mathcal{A}$ is said to be adversarial for f at node v w.r.t. the reference classifier g if the following conditions are satisfied:*

$$\begin{aligned}
&\textit{i.} \quad f(\mathbf{X}, \mathbf{A}, y_{trn})_v = g(\mathbf{X}, \mathbf{A}, y_{trn})_v && \textit{(correct clean prediction)} \\
&\textit{ii.} \quad g(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_{trn})_v = g(\mathbf{X}, \mathbf{A}, y_{trn})_v && \textit{(perturbation preserves semantics)} \\
&\textit{iii.} \quad f(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_{trn})_v \neq g(\mathbf{X}, \mathbf{A}, y_{trn})_v && \textit{(node classifier changes prediction)}
\end{aligned}$$

Definition 1 says that a perturbed graph $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in \mathcal{B}(\mathbf{X}, \mathbf{A})$ only then is adversarial, if $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})$ does not only change the prediction of the node classifier f (iii), but also lets the original label unchanged (ii). The first constraint (i) stems from the fact that if f and g disagree on the clean graph at node v this should represent a case of misclassification captured by standard error metrics such as accuracy.

Suggala et al. (2019) use the concept of a reference classifier to, in similar spirit, define semantics-aware adversarial perturbations for i.i.d. data, with a focus on the image domain. However, what has not been considered so far, is that the reference classifier allows us to characterize the exact opposite behaviour of an adversarial example: if a classifier f does not change its prediction for a perturbed graph $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})$ even though the semantic content has changed. As this would mean that f is robust beyond the point of semantic change, we call this behaviour *over-robustness*:

Definition 2. *Let f be a node-classifier and g a reference classifier. Then the perturbed graph $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in \mathcal{B}(\mathbf{X}, \mathbf{A})$ chosen by an adversary $\mathcal{A}$ is said to be an over-robust example for f at node v w.r.t. the reference classifier g if the following conditions are satisfied:*

$$\textit{i.} \quad f(\mathbf{X}, \mathbf{A}, y_{trn})_v = g(\mathbf{X}, \mathbf{A}, y_{trn})_v \qquad \textit{(correct clean prediction)}$$

$$\textit{ii.} \quad g(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_{trn})_v \neq g(\mathbf{X}, \mathbf{A}, y_{trn})_v \qquad \textit{(perturbation changes semantics)}$$

$$\textit{iii.} \quad f(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}, y_{trn})_v = g(\mathbf{X}, \mathbf{A}, y_{trn})_v \qquad \textit{(node classifier stays unchanged)}$$

If there exists such an over-robust example, we call f over-robust at node v w.r.t. g.

Definition 2 may be of particular interest in the graph domain, where perturbation sets $\mathcal{B}(\mathbf{X}, \mathbf{A})$ often include graphs $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})$ which allow significant changes to the neighbourhood structure of v, but do not allow easy manual content inspections. Indeed, Section 5 shows that all assessed GNNs are over-robust for common choices of $\mathcal{B}(\mathbf{X}, \mathbf{A})$ for many test nodes v in CSBM graphs. This may not be as relevant in the image domain, where small ℓ_p-norm perturbations are visually imperceptible.

In the following, we develop a deeper understanding of over-robustness in contrast to an adversarial example (Definition 1). Denote $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ and collect all over-robust examples in a set $\mathcal{B}_O(\mathcal{G}, v) \subset \mathcal{B}(\mathcal{G})$ and adversarial examples in a set $\mathcal{B}_A(\mathcal{G}, v) \subset \mathcal{B}(\mathcal{G})$. Then, following from Definition 1 and 2, the set of over-robust examples $\mathcal{B}_O(\mathcal{G}, v)$ and the set of adversarial examples $\mathcal{B}_A(\mathcal{G}, v)$ are disjoint. We also find that in general, for a given over-robust example $\tilde{\mathcal{G}} \in \mathcal{B}(\mathcal{G})$ one can't always find a corresponding clean graph $\mathcal{G}' \neq \mathcal{G}$ not in $\mathcal{B}(\mathcal{G})$ for which $\tilde{\mathcal{G}}$ is an adversarial example.

Proposition 1. *Given f is a constant classifier, then $\mathcal{B}_A(\mathcal{G}, v)$ is empty for every possible graph $\mathcal{G}$.*

This follows as f can never fulfill both, item (i) and item (iii) in Definition 1. However, over-robust examples for f may exists. Every example in $\mathcal{B}(\mathcal{G})$ will be over-robust, for which g changes its label for node v. This shows that *over-robustness* can differentiate two classifiers, which have the same adversarial robustness, but one has learned a better decision boundary, while the other has not.

Now, we look at a non-constant classifier with a well-defined decision boundary in input space[1]. In Figure 2a the decision boundary of a classifier f follows the one of a base classifier g except for the dotted line. The dashed region between f's and g's decision boundary is a region of over-robustness for the blue class and a region of adversarial examples for the red class. In practice, the number of perturbations and hence, the extent of the perturbation sets $\mathcal{B}(\cdot)$ is bounded (see Figure 2b). As a result, using adversarial examples, it is only possible to measure the right boundary of the dashed area. The concept of over-robustness allows us to additionally measure the left boundary and hence, provides us with a more complete picture of the robustness of f. We further develop other interesting conceptual cases in Appendix A.

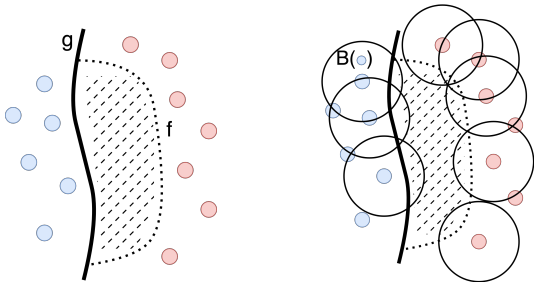


Figure 2: Conceptual differences between over- and adversarial robustness. a) The decision boundary of classifier f follows the one of a base classifier g except for the dotted line. b) Finite perturbation sets $\mathcal{B}(\cdot)$ intersect only from one side with the dashed area.

4 Bayes Classifier as Reference Classifier

In the following, we derive a Bayes optimal classifier to use as a reference classifier g. Using the Bayes decision rule as g is a natural choice, because it provides us with the information, if another class is now more likely based on the true data generating distribution and hence, is the closest we can get to a semantic content returning oracle Ω (see Section 3).

Assume that we want to learn an inductive node-classifier f on a given, fully labeled graph $(\mathbf{X}, \mathbf{A}, y) \sim \mathcal{D}_n$ with n nodes. We will focus on the most simple case of classifying an induc-

[1]This is not true e.g. for random functions. However, it is for most models and in particular GNNs.

tively sampled node. We denote the conditional distribution over graphs with an inductively added node as $\mathcal{D}(\mathbf{X}, \mathbf{A}, y)$. Then, the target node v corresponds to the newly sampled, $n+1$-th node. How well our classifier f generalizes to the newly added node is captured by the expected $0/1$-loss of f:

$$\mathop{\mathbb{E}}_{(\mathbf{X}', \mathbf{A}', y') \sim \mathcal{D}(\mathbf{X}, \mathbf{A}, y)} [\ell_{0/1}(y'_v, f(\mathbf{X}', \mathbf{A}', y)_v] \tag{1}$$

To derive a Bayes optimality, we have to find an optimal classifier f^* for v, depending on $(\mathbf{X}', \mathbf{A}', y)$, minimizing (1). The following theorem shows that, similar to inductive classification for i.i.d. data, f^* should choose the most likely class based on the seen data (Proof in Appendix B.1):

Theorem 1. *The Bayes optimal classifier, minimizing the expected $0/1$-loss (1), is $f^*(\mathbf{X}', \mathbf{A}', y)_v = \arg\max_{\hat{y} \in \{0,1\}} \mathbb{P}[y'_v = \hat{y} | \mathbf{X}', \mathbf{A}', y]$.*

4.1 Robustness-Accuracy Tradeoff

We show that given a Bayes optimal reference classifier g, our robustness notions (Definition 1 and 2) imply that optimal robustness, i.e. the non-existence of adversarial and over-robust examples for f, is possible while preserving good generalization in the sense of (1). Our argumentation for the non-i.i.d. graph data case takes inspiration from Suggala et al. (2019)'s study for i.i.d. data. In the following, we assume, we are given a graph $\mathcal{G}'$ sampled from $\mathcal{D}(\mathcal{G}, y)$, where $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ represents a fully labeled y training graph with n nodes. Let v refer to the $n+1$-th (inductively added) node. First, observe from Definition 1 that a graph $\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}')$ is an adversarial example, iff $g(\tilde{\mathcal{G}}', y)_v = g(\mathcal{G}', y)_v$ and $\ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\mathcal{G}', y)_v) - \ell_{0/1}(f(\mathcal{G}', y)_v, g(\mathcal{G}', y)_v) = 1$. In analogy to the generalization error (1), we define the adversarial generalization error:

Definition 3. *Let f be a node classifier and g a reference node classifier. Then the expected adversarial $0/1$-loss for an inductively added target node v is defined as the fraction of conditionally sampled graphs which can be adversarially perturbed:*

$$\mathop{\mathbb{E}}_{(\mathcal{G}', y') \sim \mathcal{D}(\mathcal{G}, y)} \left[\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ g(\tilde{\mathcal{G}}', y)_v = g(\mathcal{G}', y)_v}} \ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\mathcal{G}', y)_v) - \ell_{0/1}(f(\mathcal{G}', y)_v, g(\mathcal{G}', y)_v) \right] \tag{2}$$

From Definition 2 it follows that a graph $\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G})$ is an over-robust example, iff $f(\tilde{\mathcal{G}}', y)_v = f(\mathcal{G}', y)_v$ and $\ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\mathcal{G}', y)_v) - \ell_{0/1}(g(\mathcal{G}', y)_v, f(\mathcal{G}', y)_v) = 1$. Thus, we define:

Definition 4. *Let f be a node classifier and g a base node classifier. Then the expected over-robust $0/1$-loss for an inductively added target node v is defined as the fraction of resulting graphs which are over-robust examples:*

$$\mathop{\mathbb{E}}_{(\mathcal{G}', y') \sim \mathcal{D}(\mathcal{G}, y)} \left[\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ f(\tilde{\mathcal{G}}, y)_v = f(\mathcal{G}', y)_v}} \ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\mathcal{G}', y)_v) - \ell_{0/1}(g(\mathcal{G}', y)_v, f(\mathcal{G}', y)_v) \right] \tag{3}$$

We denote the expected adversarial $0/1$-loss as $\mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v$ and the expected over-robust $0/1$-loss as $\mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v$. Minimizing only one of the robust objectives and disregarding the standard loss (1), may not yield a sensible classifier. Exemplary, the adversarial loss (2) achieves its minimal value of 0 for a constant classifier. Therefore, we collect them in an overall expected robustness loss term:

$$\mathscr{L}^{rob}_{\mathcal{G},y}(f, g)_v = \lambda_1 \mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v + \lambda_2 \mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v \tag{4}$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ define how much weight we give to the adversarial and the over-robust loss. Now, we want to find a node-classifier f with small robust and standard loss. Denoting the expected $0/1$-loss (1) as $\mathscr{L}_{\mathcal{G},y}(f)_v$, this leads us to optimize the following objective:

$$\arg\min_{f \in \mathcal{H}} \mathscr{L}_{\mathcal{G},y}(f)_v + \lambda \mathscr{L}^{rob}_{\mathcal{G},y}(f, g)_v \tag{5}$$

where $\lambda \geq 0$ defines a tradeoff between standard accuracy and robustness and $\mathcal{H}$ represent a set of admissible functions, e.g. defined by a chosen class of GNNs. Now, the following holds:

Theorem 2. *Assume a set of admissible functions $\mathcal{H}$, which includes a Bayes optimal classifier f^*_{Bayes} and let the reference classifier g be itself a Bayes optimal classifier. Then, any minimizer $f^* \in \mathcal{H}$ of (5) is an optimal Bayes classifier.*

Proof see Appendix B.2.1. Theorem 2 implies that minimizing both, the standard and robust loss for any $\lambda \geq 0$, always yields a Bayes optimal classifier. Therefore, optimizing for robustness does not tradeoff accuracy of the found classifier by (5) and hence, establishes that classifying an inductively sampled node does not suffer from a robustness-accuracy tradeoff. Theorem 2 raises the important question if common GNNs define function classes $\mathcal{H}_{GNN}$ expressive enough to represent a Bayes classifier for (1) and hence, can achieve optimal robustness. Theorem 3 in Appendix B.2.2 shows that only being a minimizer for the robust loss $\mathscr{L}^{rob}_{\mathcal{G},y}(f,g)_v$ does not imply good generalization.

5 RESULTS

Using Contextual Stochastic Block Models (CSBMs) we measure the extent of semantic content violations in common threat models (Section 5.1). Then, we study over-robustness in CSBMs (Section 5.2) and real-world graphs (Section 5.2.1). In CSBMs, we use the Bayes optimal classifier (Theorem 1), denoted g, to measure semantic change. The robustness of the Bayes classifier defines the maximal meaningful robustness achievable (see Section 3).

Experimental Setup. We sample training graphs with $n = 1000$ nodes from a $\text{CSBM}^{\mu,\sigma^2}_{n,p,q}$ in the hard regime (Section 2). Each element of the class mean vector $\mu \in \mathbb{R}^d$ is set to $K\sigma/2\sqrt{d}$, resulting in a distance between the class means of $K\sigma$. We set $\sigma = 1$ and vary K from close to *no* discriminative features $K = 0.1$ to making structure information *unnecessary* $K = 5$ (see Table 1). We choose $p = 0.63\%$ and $q = 0.15\%$ resulting in the expected number of same-class and different-class edges for a given node to fit CORA (Sen et al., 2008). Following Fountoulakis et al. (2022), we set $d = \lfloor n/\ln^2(n) \rceil = 21$. We use an $80\%/20\%$ train/validation split on the nodes. As usual in the inductive setting, we remove the validation nodes from the graph during training. At test time, we inductively sample 1000 times an additional node conditioned on the training graph. For each K, we sample 10 different training graphs.

Accuracy (Bayes)		
K	$\mathbf{X}$	$(\mathbf{X}, \mathbf{A})$
0.1	50.8%	89.7%
0.5	59.0%	90.3%
1.0	68.4%	91.7%
1.5	76.5%	93.1%
2.0	83.4%	94.7%
3.0	92.6%	97.4%
4.0	97.5%	99.0%
5.0	99.3%	99.8%

Table 1: Mean accuracy of the Bayes optimal classifier on test nodes v with $(\mathbf{X}, \mathbf{A})$ and without $(\mathbf{X})$ structure information.

Models and Attacks. We study a wide range of popular GNN architectures: Graph Convolutional Networks (GCN) (Kipf & Welling, 2017), Simplified Graph Convolutions (SGC) (Wu et al., 2019a), Graph Attention Networks (GAT) (Veličković et al., 2018), GATv2 (Brody et al., 2022), APPNP (**?**), and GraphSAGE (Hamilton et al., 2017). Furthermore, we study a simple Multi-Layer Perceptron (MLP) and Label Propagation (LP) (Zhou et al., 2004). The combination of a model with LP (Huang et al., 2021) is denoted by *Model+LP* (see Section 2). To find adversarial examples, we employ the established attacks *Nettack* (Zügner et al., 2018) and *DICE* (random addition of different-class edges) (Waniek et al., 2018). To find over-robust examples, we use a "weak" attack which we call ℓ_2-*weak*: Connect to the closest different-class nodes in ℓ_2-norm. Theorem 4 in Appendix B.3 shows that a strategy to change, with least structure changes, the true most likely class on CSBMs, i.e. an "optimal attack" against the Bayes classifier g, is given by (arbitrarily) disconnecting same-class edges and adding different-class edges to the target node. Therefore, the attacks *DICE* and ℓ_2-*weak* have the same effect on the semantic content of a graph. We investigate varying local budgets Δ from 1 up to the degree (deg) of a node + 2, similarly to Zügner et al. (2018). We call the induced perturbation set $\mathcal{B}_\Delta(\cdot)$. Further details, including the hyperparameter settings can be found in Appendix D.

Robustness Metrics. To analyse the robustness of varying models across different graphs, we need to develop comparable metrics summarizing the robustness properties of a model f on a given graph. First, to correct for the different degrees of nodes, we measure the adversarial robustness of f (w.r.t. g) at node v relative to v's degree and average over all test nodes V'[2]:

$$R(f,g) = \frac{1}{|V'|} \sum_{v \in V'} \frac{\text{Robustness}(f,g,v)}{\deg(v)} \tag{6}$$

where g represents the Bayes classifier and hence, $\text{Robustness}(f,g,v)$ refers to the (minimal) number of semantics-aware structure changes f is robust against (Definition 1). Exemplary,

[2]Excluding degree 0 nodes. This is one limitation of this metric, however, these are very rare in the generated CSBM graphs and non-existing in common benchmark datasets such as CORA.

$R(f, g) = 0.5$ would mean that on average, node predictions are robust against changing 50% of the neighbourhood structure. Using the true labels y instead of a reference classifier g in (6), yields the conventional (degree-corrected) adversarial robustness, unaware of semantic change, which we denote $R(f) := R(f, y)$. To measure **over-robustness**, we measure the fraction of conventional adversarial robustness $R(f)$, which cannot be explained by semantic-preserving robustness $R(f, g)$: $R^{over} = 1 - R(f, g)/R(f)$. Exemplary, $R^{over} = 0.2$ means that 20% of the measured robustness is robustness beyond semantic change. In Appendix C we present a metric for semantic-aware adversarial robustness and how to calculate an overall robustness measure using both metrics.

5.1 Extent of Semantic Content Change in Common Perturbation Models

We investigate how prevalent perturbed graphs with changed semantic content are for common perturbation model choices. We denote the perturbation set allowing a local budget of Δ edge perturbations as $\mathcal{B}_\Delta(\cdot)$. Table 2 shows the fraction of test nodes, for which we find perturbed graphs in $\mathcal{B}_\Delta(\cdot)$ with changed ground truth labels. Surprisingly, even for very modest budgets, if structure matters ($K \leq 3$), this fraction is significant. Exemplary, for K=1.0 and $\mathcal{B}_{\text{deg+2}}(\cdot)$, we find perturbed graphs with changed semantic content for 99.4% of the target nodes. This establishes for CSBMs, a *negative* answer to a question formulated in the introduction: *If structure matters, does completely reconnecting a node preserve its semantic content?* Similar to CSBMs, nodes in real-world graphs have mainly low-degree (Figure 8 in Appendix E.1). This provides evidence that similar conclusion could be drawn for certain real-world graphs. The examined $\mathcal{B}_\Delta(\cdot)$ subsume all threat models against edge-perturbations employing the ℓ_0-norm to measure small changes to the graph's structure, as we investigate the lowest choices of local budgets possible[3] and for these already find a large percentage of perturbed graphs, violating the semantics-preservation assumption.

Threat Models	**K** 0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
$\mathcal{B}_1(\cdot)$	14.3	11.2	9.1	6.8	4.4	1.9	0.8	0.2
$\mathcal{B}_2(\cdot)$	35.9	31.2	25.7	19.8	14.1	6.2	2.2	0.7
$\mathcal{B}_3(\cdot)$	58.5	53.8	46.8	38.2	28.8	14.3	5.1	1.7
$\mathcal{B}_4(\cdot)$	76.5	73.0	66.6	58.1	47.0	25.7	9.8	3.4
$\mathcal{B}_{\text{deg}}(\cdot)$	75.7	60.0	55.4	49.1	39.6	21.9	9.0	3.2
$\mathcal{B}_{\text{deg+2}}(\cdot)$	100	100	99.4	92.9	80.5	51.7	24.8	9.1

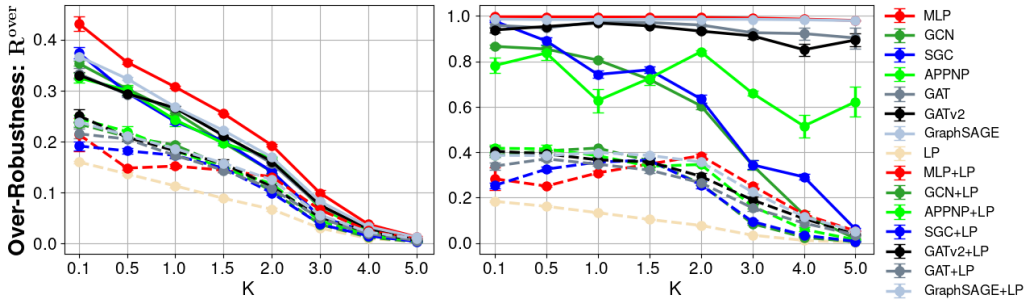
Table 2: Percentage (%) of test nodes for which perturbed graphs in $\mathcal{B}_\Delta(\cdot)$ violate semantic content preservation, i.e. have changed ground truth labels. Calculated by connecting Δ different class nodes (ℓ_2-weak) to every target node. For K=4.0 and K=5.0 structure is not necessary for good generalization (Table 1). Results using DICE and Nettack as search-heuristics are similar (see Appendix E.3). Standard deviations are insignificant and hence, omitted for brevity.

Values in Table 2 are lower bounds on the prevalence of graphs in $\mathcal{B}_\Delta(\cdot)$ with changed semantic content. We calculate these values by connecting Δ different-class nodes (ℓ_2-weak) to every target node and hence, the constructed perturbed graphs $\tilde{\mathcal{G}}$ are at the boundary of $\mathcal{B}_\Delta(\cdot)$. Then, we count how many $\tilde{\mathcal{G}}$ have changed their true most likely class using the Bayes classifier g. Thus, exemplary, if a classifier f is robust against $\mathcal{B}_3(\cdot)$ for CSBMs with K=1.0, f classifies the found graphs at the boundary of $\mathcal{B}_3(\cdot)$ wrong in 47% of cases. Therefore, f shows high **over-robustness** R^{over}.

5.2 Over-Robustness of Graph Neural Networks

Section 5.1 establishes the **existence** of perturbed graphs with changed semantic content in common threat models. Now, we examine how much of the measured robustness of common GNNs can actually be attributed to over-robustness (R^{over}), i.e. to robustness beyond semantic change. As a qualitative example, we study a local budget of the degree of the target nodes $\mathcal{B}_{\text{deg}}(\cdot)$. Figure 3a shows the over-robustness of GNNs when attacking their classification of inductively added nodes. For $K \leq 3$ the graph structure is relevant in the prediction (see Table 1). We find that in this regime, a significant amount of the measured robustness of **all GNNs** can be attributed to **over-robustness**. Exemplary, 30.3% of the conventional adversarial robustness measurement of a GCN for $K = 0.5$ turns out to be over-robustness. Label propagation achieves the lowest R^{over} and R^{over} **significantly reduces** when LP is applied on top of a GNN. Exemplary, GCN+LP for $K = 0.5$ drops

[3] Global budgets again result in local edge-perturbations, however, now commonly allowing for way stronger perturbations than $\mathcal{B}_{\text{deg+2}}(\cdot)$ to individual nodes. The number of allowed perturbation is usually set to a small one or two digit percentage of the total number of edges in the graph. For CORA with 5278 edges, a relative budget of 5% leads to a budget of 263 edge-changes, which can be distributed in the graph without restriction.



(a) Local Budget Δ**:** Degree of Node (b) No Budget Restriction.

Figure 3: Fraction of Robustness beyond Semantic Change (Attack: ℓ_2-weak). Dashed Lines refer to LP. (a) A large part of the measured robustness against $\mathcal{B}_{\text{deg}}(\cdot)$ (ℓ_2-weak) can be attributed to over-robustness. (b) Applying ℓ_2-weak without budget restriction until it changes a classifiers prediction. Note that over-robustness can be significantly reduced by label propagation and for (b) that some over-robust models, especially for high K, show high adversarial robustness (Appendix E.5).

to $R^{over} = 20.9\%$. Adding LP does **not** decrease test-accuracy (see Figure 9 in Appendix E.2) and often **increases** adversarial robustness as long as structure matters (see Figure 10 in Appendix E.5.1).

An MLP by achieving maximal adversarial robustness provides an estimate of the upper bound on the over-robustness for a particular K. Exemplary, it has 43% over-robustness for $K = 0.1$. This means that for a perfectly robust classifier against $\mathcal{B}_{\text{deg}}(\cdot)$ for CSBM graphs with $K = 0.1$, we can expect 43% of conventional adversarial robustness to be undesirable over-robustness. Note that all GNNs are **close** to this upper bound. Stronger attacks such as Nettack also show that a significant part of the measured robustness is over-robustness (see Appendix E.4). Exemplary, Nettack performs strongest against SGC and GCN. However, for a GCN at $K = 0.5$, still 11.4% of the measured robustness is in fact over-robustness. An MLP for Nettack for $K = 2$ still shows 19.2% R^{over}, indicating that we can expect a model robust against Nettack in $\mathcal{B}_{\text{deg}}(\cdot)$ to have high R^{over}. For bounded perturbation sets $\mathcal{B}_{\Delta}(\cdot)$, maximal over-robustness necessarily decreases if K increases, as the more informative features are, the more structure changes it takes to change the semantic content (i.e. the Bayes decision). Thus, less graphs in $\mathcal{B}_{\Delta}(\cdot)$ have changed semantics (also see Table 2).

For Figure 3b we apply ℓ_2-weak without budget restriction until it changes a model's prediction. As a result, we find that all GNNs have extensive areas of over-robustness in input space (compare to Figure 5 in Appendix A). We find that for high K, some GNNs have high (semantics-aware) adversarial robustness (Figure 10 in Appendix E.5.1) additional to high R^{over}. Using the harmonic mean of both, we find the overall robustness-rankings of GNNs can change compared to only using conventional adversarial robustness (see Figure 11 in Appendix E.5.1). However, only considering (semantics-aware) adversarial robustness, the ranking is approximately preserved. This is good news for evaluating adversarial robustness of GNNs on real-world graphs, as high conventional adversarial robustness can to translate to high semantics-aware adversarial robustness on CSBMs. Figure 12 in Appendix E.5.1 shows that the best harmonic mean robustness is achieved by MLP+LP, which also significantly leads regarding adversarial robustness, while achieving competitive test-accuracies.

5.2.1 Over-Robustness on Real World Graphs

Table 2 shows that only a few perturbations can change the semantic-content a graph encodes about a target node. Additionally, if structure matters ($K \leq 3$), the Bayes decision on CSBMs changes on average after at most changing as many edges as the target node's degree. This is visualized in Figure 1 measured for K=1.5 and for other K in Figure 17 in Appendix E.5.4. It is challenging to derive a reference classifier for real-world datasets and hence, directly measure over-robustness. However, we can investigate the degree-dependent robustness of GNNs and see if we similarly find high robustness beyond the degree of nodes. Figure 4 shows that a majority of test node predictions of a GCN on (inductive) Cora-ML (Bojchevski & Günnemann, 2017) are robust beyond their degree, by several multiples. The median robustness for degree 1 nodes, lies at over 10 structure changes. Figure 16 in Appendix E.5.3 shows a similar plot for CSBMs. Results are obtained by applying a

variant of ℓ_2-weak on a target node v. However, as Cora-ML is a multi-class dataset, we ensure all inserted edges connect to the same class c, which is different to the class of v. The vulnerability of a GCN to adversarial attacks likely stems from the lower quartile of robust node classifications in Figure 4. We conjecture *over-robustness* for the upper quartile of highly robust node classifications. In Appendix E.5.5 we show that combining the GCN with LP significantly reduces the upper extent of its robustness while achieving similar test accuracy and show similar results on Citeseer (Sen et al., 2008).

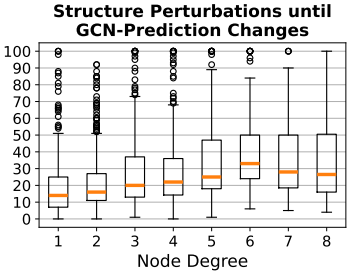


Figure 4: (Inductive) Cora-ML.

6 Related Work

Due to the large body of related work, we only discuss the most relevant here and refer for an extended discussion to Appendix F. The problem of semantic content preservation was theoretically studied by Suggala et al. (2019). Similarly, their work shows that there is no robustness-accuracy tradeoff, postulated among others by Tsipras et al. (2019), for semantics-aware adversarial robustness. However, they only discuss i.i.d. data and focus on the image domain. To the best of our knowledge, we are the first to study an (adversarial) notion of over-robustness, and a robustness-accuracy tradeoff for non-i.i.d. data. For graphs, Dai et al. (2018) note the possibility of measuring semantic content with a gold standard classifier and generate semantic-preserving perturbations for graph-classification on Erdős-Rényi graphs. Although no work has explicitly addressed semantic content preservation for node classification, we list works going beyond ℓ_2-norm restrictions: Zügner et al. (2018) proposed to approximate unnoticeability by preserving a (power-law) degree distribution; Li et al. (2021) introduced a metric for degree assortativity, but did not restrict perturbations; Chen et al. (2022) propose other homophily metrics for unnoticability, but focus on different perturbations to the graph, namely adding malicious nodes.

7 Discussion and Conclusion

We have shown that common threat models on CSBMs include many graphs with changed semantic content. But, we also found that the same threat models include truly adversarial examples. Thus, it needs two robustness notions, adversarial and over-robustness for a complete picture in graph machine learning. In CSBMs, this dichotomy is caused by the low-degrees of nodes and the brittleness of their class-membership to a few edges. We have shown that for CSBMs (full) conventional robustness against common threat models would lead to sub-optimal generalization and large robustness *beyond* the point of semantic change. As real-world graphs also contain mainly low-degree nodes, we think this calls for more caution when applying ℓ_0-norm restricted threat models. We think these threat models should not be an end to, but the beginning of an investigation into realistic perturbations models and we see works thinking about unnoticeability as positive directions into this endeavour. We have shown that on CSBMs, a significant part of conventional robustness of GNNs can be attributed to over-robustness, with similar patterns on real-world graphs. This raises the question, what kind of robustness do defenses improve on in GNNs? As a positive indication, we have found that in CSBMs conventional adversarial robustness translates to semantics-aware adversarial robustness. However, including over-robustness can change robustness rankings.

Applying label propagation on top of GNN predictions has shown to be a simple way to significantly reduce over-robustness while not harming generalization or adversarial robustness. Therefore, label propagation can be seen as a defense against an attack, where the adversary overtakes a clean node (e.g. social media user) and, with its malicious activity, tries to stay undetected. This shows that not including the known labels in their predictions can be a significant limitation of GNNs. As visually inspecting graphs is difficult, we have shown that synthetic graph generation models can be used as important tools to further a principled understanding of graph attacks and defenses. We believe our work outlines a framework for others to build upon. Concluding, using semantics-aware robustness, we show for inductively classifying a newly sampled node, optimal robustness can be achieved while maintaining high accuracy. This is positive evidence that both robust and generalizing graph models are attainable.

REPRODUCIBILITY STATEMENT

We will release the source code and all experiment configuration files of all our experiments. Furthermore, we detail our hyperparameter search procedure and all searched through values of all our models and attacks in Appendix D. Details on the parametrization of the used CSBMs can be found in Section 5. We performed all experiments trying to control the randomness as much as possible. We set random seeds and ensure no outlier phenomena by averaging over multiple seeds (including generating multiple CSBM graphs for each CSBM parameterization) as detailed in Section 5 and Appendix D. The experimental setup of our real-world graph experiments is outlined in Appendix D.1.

ETHICS STATEMENT

Robustness is an important research direction for the reliability of machine learning in real world applications. A rigorous study counteracts possible exploits of real-world adversaries. We think that the benefits of our work outweigh the risks by and see no direct negative implications. However, there remains the possibility of non-benign usage. Specifically, perturbations that are over-robust are likely less noticeable in comparison to prior work. To mitigate this risk and other threats originating from unrobustness, we urge practitioners to assess their model's robustness (at best trying to include domain knowledge to go beyond ℓ_0-norm perturbations and closer to truly realistic threat models).

REFERENCES

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *ICML*, pp. 684–693, 2021.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. 2017.

Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1003–1013, 2020.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Jinyin Chen, Yangyang Wu, Xuanheng Xu, Yixian Chen, Haibin Zheng, and Qi Xuan. Fast gradient attack on network embedding. *ArXiv*, abs/1809.02797, 2018.

Yongqiang Chen, Han Yang, Yonggang Zhang, MA KAILI, Tongliang Liu, Bo Han, and James Cheng. Understanding and improving graph injection attack by promoting unnoticeability. In *International Conference on Learning Representations*, 2022.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1115–1124. PMLR, 10–15 Jul 2018.

Yash Deshpande, Andrea Montanari, Elchanan Mossel, and Subhabrata Sen. Contextual stochastic block models. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 8590–8602, Red Hook, NY, USA, 2018. Curran Associates Inc.

Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. All you need is low (rank): Defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 169–177, New York, NY, USA, 2020. Association for Computing Machinery.

Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. *arXiv preprint arXiv:2202.13060*, 2022.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019.

Simon Geisler, Daniel Zügner, and Stephan Günnemann. Reliable graph neural networks via robust aggregation. In *Neural Information Processing Systems, NeurIPS*, 2020.

Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 7637–7649. Curran Associates, Inc., 2021.

Simon Geisler, Johanna Sommer, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Generalization of neural combinatorial solvers through the lens of adversarial robustness. In *International Conference on Learning Representations*, 2022.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Stephan Günnemann. *Graph Neural Networks: Adversarial Robustness*, pp. 149–176. Springer Nature Singapore, Singapore, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2 edition, 2009.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved Problems in ML Safety. *arXiv e-prints*, art. arXiv:2109.13916, September 2021.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations*, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Jintang Li, Tao Xie, Chen Liang, Fenfang Xie, Xiangnan He, and Zibin Zheng. Adversarial attack on large scale graph. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1, 05 2021.

John Palowitch, Anton Tsitsulin, Brandon Mayer, and Bryan Perozzi. Graphworld: Fake graphs bring real insights for gnns. KDD '22, pp. 3691–3701, New York, NY, USA, 2022. Association for Computing Machinery.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016.

Jan Schuchardt, Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Collective robustness certificates: Exploiting interdependence in graph neural networks. In *International Conference on Learning Representations*, 2021.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data articles. *AI Magazine*, 29:93–106, 09 2008.

Arun Sai Suggala, Adarsh Prasad, Vaishnavh Nagarajan, and Pradeep Ravikumar. Revisiting adversarial risk. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2331–2339. PMLR, 16–18 Apr 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Marcin Waniek, Tomasz Michalak, Talal Rahwan, and Michael Wooldridge. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2, 02 2018.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 09–15 Jun 2019a.

Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4816–4823. International Joint Conferences on Artificial Intelligence Organization, 7 2019b.

Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 3961–3967. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

Kaidi Xu, Sijia Liu, Pin-Yu Chen, Mengshu Sun, Caiwen Ding, Bhavya Kailkhura, and Xue Lin. Towards an efficient and general framework of robust training for graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8479–8483, 2020.

Qinkai Zheng, Xu Zou, Yuxiao Dong, Yukuo Cen, Da Yin, Jiarong Xu, Yang Yang, and Jie Tang. Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning. *Neural Information Processing Systems Track on Datasets and Benchmarks 2021*, 2021.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf (eds.), *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pp. 321–328. MIT Press, 2004.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1399–1407, New York, NY, USA, 2019. Association for Computing Machinery.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations (ICLR)*, 2019a.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining*, New York, NY, USA, 2019b. ACM.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *SIGKDD*, pp. 2847–2856, 2018.

A Conceptual Differences between Over- and Adversarial Robustness

Figure 5a shows the decision boundary of a classifier f following the one of a base classifier g except for the dotted line. The dashed region between f's and g's decision boundary is a region of over-robustness for the blue class and a region of adversarial examples for the red class.

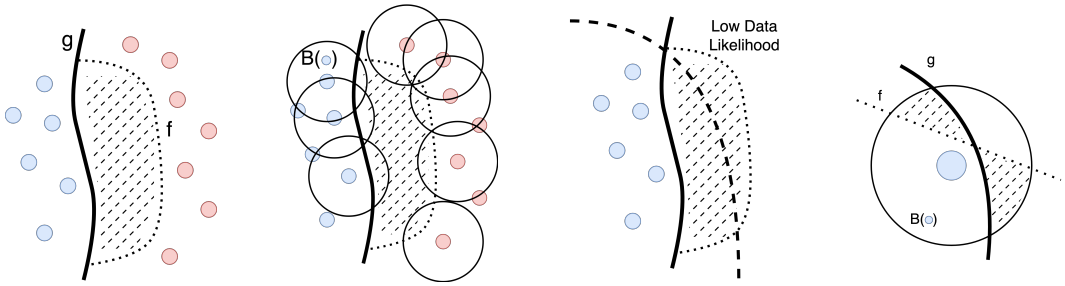


Figure 5: Conceptual differences between over- and adversarial robustness. a) The decision boundary of classifier f follows the one of a base classifier g except for the dotted line. b) Finite perturbation budgets induce bounded perturbation sets $\mathcal{B}(\cdot)$ intersecting only from one side with the dashed area. c) The red class is not seen because it lies in a low data likelihood region. d) Zoomed: A node whose perturbation set includes a region of adversarial and over-robust examples.

Note that in Figure 5b, using the classical concept of adversarial robustness, perturbed examples of blue datapoints crossing the decision boundary of g but still in $\mathcal{B}(\cdot)$ will be judged adversarially robust. Therefore, they may be used to provide a learning signal for f to further solidify its too insensitive decision boundary. Using our refined concept of adversarial robustness, the set of potentially adversarial examples $\mathcal{B}(\cdot)$ is (correctly) cut off at the decision boundary of g.

In Figure 2a it is assumed that datapoints on both sides of the decision boundary g have been sampled. This may be likely if every datapoint in the input space has a comparable sampling probability. However, in practice there are regions of high and low data likelihood and hence, it could be that datapoints on one side of the dashed regions have not been sampled as exemplified in Figure 5c). There, the dashed line indicates the transition from a high to low data likelihood area. As a result, only the concept of over-robustness can capture the misbehaviour of the classifier f. The reverse scenario is also possible in which only the examples of the right class are sampled and the left class is in a low likelihood region. Indeed, in our results (see Section 5.2), there are some cases where we measure both high adversarial and over-robustness, exactly fitting the scenario visualized in Figure 5c. Note that it makes sense to robustify a classifier even against low-likelihood events as in safety-critical scenarios correct behaviour for unusual or rare events is crucial (Hendrycks et al., 2021).

Figure 5d zooms in to a more intricate case. Disregarding the base classifier g would lead to wrongly interpreting every example above the decision boundary of f as adversarial. With our refined notions, it is possible correctly identify the adversarial region as above f until g, the over-robust region as below the decision boundary of f but right of g and the correctly classified area in the top-right.

B PROOFS

B.1 BAYES CLASSIFIER

We proof Theorem 1 by deriving the base classifier for multi-class node classification with C classes. Theorem 1 then follows as special case by setting $C = 2$. We restate Theorem 1 for multiple classes:

Theorem 1. *The optimal (Bayes) classifier, minimizing the expected* $0/1$*-loss (1), is* $f^*(\mathbf{X}', \mathbf{A}', y)_v = \underset{\hat{y} \in \{0,\dots,C-1\}}{argmax} \mathbb{P}[y'_v = \hat{y} | \mathbf{X}', \mathbf{A}', y]$.

Proof. Lets denote by x_{-i} all elements of a vector x except the i-th one. First, note that $\mathcal{D}(\mathbf{X}, \mathbf{A}, y)$ from which we sample $(\mathbf{X}', \mathbf{A}', y')$ defines a conditional joint distribution

$$\begin{aligned}
\mathbb{P}[\mathbf{X}', \mathbf{A}', y' | \mathbf{X}, \mathbf{A}, y] &= \mathbb{P}[y'_v | \mathbf{X}', \mathbf{A}', y'_{-v}, \mathbf{X}, \mathbf{A}, y] \cdot \mathbb{P}[\mathbf{X}', \mathbf{A}', y'_{-v} | \mathbf{X}, \mathbf{A}, y] \\
&= \mathbb{P}[y'_v | \mathbf{X}', \mathbf{A}', y] \cdot \mathbb{P}[\mathbf{X}', \mathbf{A}', y | \mathbf{X}, \mathbf{A}, y] \\
&= \mathbb{P}[y'_v | \mathbf{X}', \mathbf{A}', y] \cdot \mathbb{P}[\mathbf{X}', \mathbf{A}' | \mathbf{X}, \mathbf{A}, y] \qquad (7)
\end{aligned}$$

where the first line follows from the basic definition of conditional probability, the second lines from the definition of the inductive sampling scheme (i.e., $y'_{-v} = y$ and similar for $\mathbf{X}'$ and $\mathbf{A}'$), and the third line from $\mathbb{P}[y | \mathbf{X}, \mathbf{A}, y] = 1$. Now, we can rewrite the expected loss (1) with respect to these probabilities as

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}', \mathbf{A}' | \mathbf{X}, \mathbf{A}, y} \left[\mathbb{E}_{y'_v | \mathbf{X}', \mathbf{A}', y} [\ell_{0/1}(y'_v, f(\mathbf{X}', \mathbf{A}', y)_v)] \right] \\
&= \mathbb{E}_{\mathbf{X}', \mathbf{A}' | \mathbf{X}, \mathbf{A}, y} \left[\sum_{k=0}^{C-1} \ell_{0/1}(k, f(\mathbf{X}', \mathbf{A}', y)_v) \cdot \mathbb{P}[y'_v = k | \mathbf{X}', \mathbf{A}', y] \right] \qquad (8)
\end{aligned}$$

The following argument is adapted from Hastie et al. (2009). Equation (8) is minimal for a classifier f^*, if it is (point-wise) minimal for every $(\mathbf{X}', \mathbf{A}', y)$. This means

$$\begin{aligned}
f^*(\mathbf{X}', \mathbf{A}', y)_v &= \underset{\hat{y} \in \{0,\dots,C-1\}}{argmin} \sum_{k=0}^{C-1} \ell_{0/1}(k, \hat{y}) \cdot \mathbb{P}[y'_v = k | \mathbf{X}', \mathbf{A}', y] \\
&= \underset{\hat{y} \in \{0,\dots,C-1\}}{argmin} \sum_{k=0}^{C-1} (1 - \mathbb{I}[k = \hat{y}]) \cdot \mathbb{P}[y'_v = k | \mathbf{X}', \mathbf{A}', y] \\
&= \underset{\hat{y} \in \{0,\dots,C-1\}}{argmax} \sum_{k=0}^{C-1} \mathbb{I}[k = \hat{y}] \cdot \mathbb{P}[y'_v = k | \mathbf{X}', \mathbf{A}', y] \qquad (9) \\
&= \underset{\hat{y} \in \{0,\dots,C-1\}}{argmax} \mathbb{P}[y'_v = \hat{y} | \mathbf{X}', \mathbf{A}', y] \qquad (10)
\end{aligned}$$

where line (10) follows from fact that in the sum of line (9), there can only be one non-zero term. Equation (10) tells us that the optimal decision is to choose the most likely class. Due to $f^*(\mathbf{X}', \mathbf{A}', y) = \underset{\hat{y} \in \{0,1\}}{argmax} \mathbb{P}[\hat{y} | \mathbf{X}', \mathbf{A}', y]$ minimizing (8) it also minimizes the expected loss (1) and hence, is an optimal Bayes classifier. □

B.2 ROBUSTNESS-ACCURACY TRADEOFF

B.2.1 THEOREM 2

For the proofs of Theorem 2 and Theorem 3, we assume that the set of cases where the two classes are equiprobable has measure zero. This is a mild assumption for instance satisfied in contextual stochastic block models.

For convenience we restate Theorem 2:

Theorem 2. *Assume a set of admissible functions $\mathcal{H}$, which includes a Bayes optimal classifier f^*_{Bayes} and let the reference classifier g be itself a Bayes optimal classifier. Then, any minimizer $f^* \in \mathcal{H}$ of (5) is an optimal Bayes classifier.*

Proof. The proof strategy is mainly adapted from Suggala et al. (2019). Let $f^*_{Bayes} \in \mathcal{H}$ and g be optimal Bayes classifiers. Assume f^* is a minimizer of (5). Further, assume that $f^*(\mathcal{G}', y)_v$ disagrees with $f^*_{Bayes}(\mathcal{G}', y)_v$ for a set of graphs $\mathcal{G}' \sim \mathcal{D}(\mathcal{G}, y)$ with non-zero measure. Note that an optimal Bayes classifier follows the decision rule $\arg\max_{\hat{y} \in \{0,1\}} \mathbb{P}[y'_v = \hat{y} | \mathcal{G}', y]$ (see Theorem 1). We assume that the set of cases where there are two or more maximal, equiprobable classes has measure zero. Then, the Bayes Decision rule is unique on the support of $\mathcal{D}(\mathcal{G}, y)$. Therefore, $f^*_{Bayes}(\mathcal{G}', y)_v = g(\mathcal{G}', y)_v$ a.e.

We will show that the joint expected loss (5) is strictly larger for f^* than f^*_{Bayes}. We start by showing that the standard expected loss $\mathscr{L}_{\mathcal{G},y}(f^*)_v > \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes})_v$:

$$
\begin{aligned}
&\mathscr{L}_{\mathcal{G},y}(f^*)_v - \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes})_v \\
&\quad = \mathop{\mathbb{E}}_{(\mathcal{G}',y') \sim \mathcal{D}(\mathcal{G},y)} [\ell_{0/1}(y'_v, f^*(\mathcal{G}', y)_v) - \ell_{0/1}(y'_v, f^*_{Bayes}(\mathcal{G}', y)_v)] \\
&\quad = \mathbb{E}_{\mathcal{G}'|\mathcal{G},y} \left[\mathbb{E}_{y'_v|\mathcal{G}',y} [\ell_{0/1}(y'_v, f^*(\mathcal{G}', y)_v) - \ell_{0/1}(y'_v, f^*_{Bayes}(\mathcal{G}', y)_v)] \right] && (11) \\
&\quad = \mathbb{E}_{\mathcal{G}'|\mathcal{G},y} \left[\mathbb{E}_{y'_v|\mathcal{G}',y} [\mathbb{I}(y'_v \neq f^*(\mathcal{G}', y)_v)] - \mathbb{E}_{y'_v|\mathcal{G}',y} [\mathbb{I}(y'_v \neq f^*_{Bayes}(\mathcal{G}', y)_v)] \right] && (12) \\
&\quad = \mathbb{E}_{\mathcal{G}'|\mathcal{G},y} \left[\mathbb{P}[y'_v \neq f^*(\mathcal{G}', y)_v | \mathcal{G}', y] - \mathbb{P}[y'_v \neq f^*_{Bayes}(\mathcal{G}', y)_v | \mathcal{G}', y] \right] && (13) \\
&\quad > 0 && (14)
\end{aligned}
$$

Line 11 follows from rewriting the conditional joint distribution defined by $\mathcal{D}(\mathcal{G}, y)$ (see Equation (7)). Line 12 follows from the definition of the $\ell_{0/1}$-loss and the linearity of expectation. The last line 14 follows from the optimal Bayes classifier being defined as a pointwise minimizer of the probability terms in line (13) and the initial assumption of $f^*(\mathcal{G}', y)_v$ disagreeing with $f^*_{Bayes}(\mathcal{G}', y)_v$ for a set of graphs with non-zero measure.

Now, we investigate the expected robust $0/1$-losses for f^* and f^*_{Bayes}. Because the decision rule defined by g equals the one of f^*_{Bayes} a.e., it follows that:

$$
\begin{aligned}
&\mathscr{L}^{adv}_{\mathcal{G},y}(f^*_{Bayes}, g)_v \\
&\quad = \mathop{\mathbb{E}}_{(\mathcal{G}',y') \sim \mathcal{D}(\mathcal{G},y)} \left[\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ g(\tilde{\mathcal{G}}', y)_v = g(\mathcal{G}', y)_v}} \ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\mathcal{G}', y)_v) - \ell_{0/1}(f(\mathcal{G}', y)_v, g(\mathcal{G}', y)_v) \right] \\
&\quad = 0
\end{aligned}
$$

and similarly:

$$
\begin{aligned}
&\mathscr{L}^{over}_{\mathcal{G},y}(f^*_{Bayes}, g)_v \\
&\quad = \mathop{\mathbb{E}}_{(\mathcal{G}',y') \sim \mathcal{D}(\mathcal{G},y)} \left[\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ f(\tilde{\mathcal{G}}, y)_v = f(\mathcal{G}', y)_v}} \ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\mathcal{G}', y)_v) - \ell_{0/1}(g(\mathcal{G}', y)_v, f(\mathcal{G}', y)_v) \right] \\
&\quad = 0
\end{aligned}
$$

Because the expected adversarial and over-robust $0/1$-losses are non-negative, i.e. $\mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v \geq 0$ and $\mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v \geq 0$ for any $f \in \mathcal{H}$, it follows that $\mathscr{L}^{rob}_{\mathcal{G},y}(f^*, g)_v \geq \mathscr{L}^{rob}_{\mathcal{G},y}(f^*_{Bayes}, g)_v$. Therefore,

$$\mathscr{L}_{\mathcal{G},y}(f^*)_v + \lambda \mathscr{L}^{rob}_{\mathcal{G},y}(f^*, g)_v > \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes}, g)_v + \lambda \mathscr{L}^{rob}_{\mathcal{G},y}(f^*_{Bayes}, g)_v$$

This is a contradiction with f^* being a minimizer of (5). Therefore, $f^*(\mathcal{G}', y)_v$ must equal $f^*_{Bayes}(\mathcal{G}', y)_v$ a.e. and hence, $f^*(\mathcal{G}', y)_v$ is an optimal Bayes classifier.

□

B.2.2 THEOREM 3

Here we investigate what happens, if we would only optimize for minimal $\mathscr{L}^{rob}_{\mathcal{G},y}(f, g)_v = \lambda_1 \mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v + \lambda_2 \mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v$. Will we also find a classifier, which has not only small robust but also small standard loss? Theorem 3 below establishes that this does not hold in general. Thus, being a minimizer for the robust loss $\mathscr{L}^{rob}_{\mathcal{G},y}(f, g)_v$, i.e., achieving optimal robustness (low over- and high adversarial robustness) does not imply achieving minimal generalization error. Theorem 3 showcases that the concepts of over- and adversarial robustness do not interchange with standard accuracy.

Theorem 3. *Assume $\mathcal{H}$ is a set of all measurable functions. Let f^*_{Bayes} be an optimal Bayes classifier and let the base classifier g also be an optimal Bayes classifier. Then, there exists a function $f^*_{rob} \in \mathcal{H}$ minimizing the robust loss $\mathscr{L}^{rob}_{\mathcal{G},y}(f^*_{rob}, g)_v$ and satisfying*

$$\mathscr{L}_{\mathcal{G},y}(f^*_{rob})_v > \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes})_v.$$

Proof. We assume that the set of cases where the two classes are equiprobable has measure zero. To prove this result, we define a node-classifier $f(\mathcal{G}', y)_v := 1 - f^*_{Bayes}(\mathcal{G}', y)$. We will first show that $f(\mathcal{G}', y)_v$ has minimal expected adversarial $0/1$-loss and later show the same for the expected over-robust $0/1$-loss.

The adversarial loss $\mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v$ takes the expectation over

$$\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ g(\tilde{\mathcal{G}}', y)_v = g(\mathcal{G}', y)_v}} \ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\mathcal{G}', y)_v) - \ell_{0/1}(f(\mathcal{G}', y)_v, g(\mathcal{G}', y)_v) \tag{15}$$

Now, note that the second term in equation (15) $\ell_{0/1}(f(\mathcal{G}', y)_v, g(\mathcal{G}', y)_v)$ is 1 by definition of f. Furthermore, because we maximize (15) over graphs with $g(\tilde{\mathcal{G}}', y)_v = g(\mathcal{G}', y)_v$, the first term $\ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\mathcal{G}', y)_v) = \ell_{0/1}(f(\tilde{\mathcal{G}}', y)_v, g(\tilde{\mathcal{G}}', y)_v)$ and hence, is always 1 by definition of f. As a result, $\mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v = 0$. Because $\mathscr{L}^{adv}_{\mathcal{G},y}(f, g)_v$ is non-negative, f achieves minimal adversarial loss.

Now we look at the expected over-robust $0/1$-loss $\mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v$. It takes the expectation over

$$\max_{\substack{\tilde{\mathcal{G}}' \in \mathcal{B}(\mathcal{G}') \\ f(\tilde{\mathcal{G}}, y)_v = f(\mathcal{G}', y)_v}} \ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\mathcal{G}', y)_v) - \ell_{0/1}(g(\mathcal{G}', y)_v, f(\mathcal{G}', y)_v) \tag{16}$$

Here again, the second term $\ell_{0/1}(g(\mathcal{G}', y)_v, f(\mathcal{G}', y)_v) = 1$ as established above. However, because in the set of graphs we are optimizing over $f(\tilde{\mathcal{G}}, y)_v = f(\mathcal{G}', y)_v$, it follows that the first term in (16) $\ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\mathcal{G}', y)_v) = \ell_{0/1}(g(\tilde{\mathcal{G}}', y)_v, f(\tilde{\mathcal{G}}', y)_v)$ and thus, by definition of f, is 1. As a result, $\mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v = 0$. Again, because $\mathscr{L}^{over}_{\mathcal{G},y}(f, g)_v$ is non-negative, f achieves minimal over-robust loss.

Therefore, we have established that f achieves optimal robustness, i.e., $\mathscr{L}^{rob}_{\mathcal{G},y}(f, y)_v = 0$.

Now we will show that $\mathscr{L}_{\mathcal{G},y}(f)_v > \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes})_v$:

$$
\begin{aligned}
&\mathscr{L}_{\mathcal{G},y}(f)_v - \mathscr{L}_{\mathcal{G},y}(f^*_{Bayes})_v \\
&\quad= \mathop{\mathbb{E}}_{(\mathcal{G}',y')\sim\mathcal{D}(\mathcal{G},y)}[\ell_{0/1}(y'_v, f(\mathcal{G}',y)_v) - \ell_{0/1}(y'_v, f^*_{Bayes}(\mathcal{G}',y)_v)] \\
&\quad= \mathop{\mathbb{E}}_{(\mathcal{G}',y')\sim\mathcal{D}(\mathcal{G},y)}[\ell_{0/1}(y'_v, 1 - f^*_{Bayes}(\mathcal{G}',y)_v) - \ell_{0/1}(y'_v, f^*_{Bayes}(\mathcal{G}',y)_v)] \\
&\quad= \mathbb{E}_{\mathcal{G}'|\mathcal{G},y}\left[\mathbb{E}_{y'_v|\mathcal{G}',y}[\mathbb{I}(y'_v \neq 1 - f^*_{Bayes}(\mathcal{G}',y)_v] - \mathbb{E}_{y'_v|\mathcal{G}',y}[\mathbb{I}(y'_v \neq f^*_{Bayes}(\mathcal{G}',y)_v)]\right] \\
&\quad= \mathbb{E}_{\mathcal{G}'|\mathcal{G},y}\left[\mathbb{E}_{y'_v|\mathcal{G}',y}[\mathbb{I}(y'_v = f^*_{Bayes}(\mathcal{G}',y)_v] - \mathbb{E}_{y'_v|\mathcal{G}',y}[\mathbb{I}(y'_v \neq f^*_{Bayes}(\mathcal{G}',y)_v)]\right] \quad (17) \\
&\quad= \mathbb{E}_{\mathcal{G}'|\mathcal{G},y}\left[\mathbb{P}[y'_v = f^*_{Bayes}(\mathcal{G}',y)_v|\mathcal{G}',y] - \mathbb{P}[y'_v \neq f^*_{Bayes}(\mathcal{G}',y)_v|\mathcal{G}',y]\right] \\
&\quad> 0
\end{aligned}
$$

Line (17) follows from the fact of binary classes. The last line follows again from the definition of the Bayes classifier and assuming that the set of cases where the two classes are equiprobable classes has measure zero.

□

B.3 Optimal Attack on CSBMs

Theorem 4. *Given a graph generated by a CSBM. The minimal number of structure changes to change the Bayes classifier (Theorem 1) for a target node v is defined by iteratively: i) connecting v to another node u with $y_v \neq y_u$ or ii) dropping a connection to another node u with $y_v = y_u$.*

Proof. Assume $(\mathbf{X}', \mathbf{A}', y') \sim \text{CSBM}^{\mu,\sigma^2}_{1,p,q}(\mathbf{X}, \mathbf{A}, y)$ with $q < p$ (homophily assumption). Recall the Bayes decision $y^* = argmax_{\hat{y}\in\{0,1\}} \mathbb{P}[y'_v = \hat{y}|\mathbf{X}', \mathbf{A}', y]$. We want to prove which structure perturbations result in a minimally changed adjacency matrix $\tilde{\mathbf{A}}'$, as measured using the ℓ_0-norm, but for which

$$
y^*_{new} = \mathop{argmax}_{\hat{y}\in\{0,1\}} \mathbb{P}[y'_v = \hat{y}|\mathbf{X}', \tilde{\mathbf{A}}', y] \neq y^* \tag{18}
$$

Therefore, we want to change

$$
\mathbb{P}[y'_v = y^*|\mathbf{X}', \mathbf{A}', y'] > \mathbb{P}[y'_v = 1 - y^*|\mathbf{X}', \mathbf{A}', y'] \tag{19}
$$

to

$$
\mathbb{P}[y'_v = y^*|\mathbf{X}', \tilde{\mathbf{A}}', y'] < \mathbb{P}[y'_v = 1 - y^*|\mathbf{X}', \tilde{\mathbf{A}}', y'] \tag{20}
$$

To achieve this, first note that we can rewrite Equation (19) using Bayes theorem:

$$
\begin{aligned}
&\frac{\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = y^*, \mathbf{X}, \mathbf{A}, y] \cdot \mathbb{P}[y'_v = y^*|\mathbf{X}, \mathbf{A}, y]}{\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|\mathbf{X}, \mathbf{A}, y]} \\
&\qquad> \frac{\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = 1 - y^*, \mathbf{X}, \mathbf{A}, y] \cdot \mathbb{P}[y'_v = 1 - y^*|\mathbf{X}, \mathbf{A}, y]}{\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|\mathbf{X}, \mathbf{A}, y]} \quad (21) \\
\iff\ &\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = y^*, \mathbf{X}, \mathbf{A}, y] > \mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = 1 - y^*, \mathbf{X}, \mathbf{A}, y] \quad (22)
\end{aligned}
$$

where in Equation 21 we use $(\mathbf{A}'_{v,:})^T = \mathbf{A}'_{:,v}$ and Equation 22 follows from $\mathbb{P}[y'_v = y^*|\mathbf{X}, \mathbf{A}, y] = \mathbb{P}[y'_v = 1 - y^*|\mathbf{X}, \mathbf{A}, y] = \frac{1}{2}$ by definition of the sampling process. Now, we take the logarithm of both sides in (22) and call the log-difference Δ:

$$
\Delta(\mathbf{A}') := \log \mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = y^*, \mathbf{X}, \mathbf{A}, y] - \log \mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:}|y'_v = 1 - y^*, \mathbf{X}, \mathbf{A}, y] \tag{23}
$$

Clearly, Equation 22 is equivalent to

$$
\Delta(\mathbf{A}') \geq 0 \tag{24}
$$

Using the properties of the sampling process of a CSBM (see Section 2), we can rewrite

$$\mathbb{P}[\mathbf{X}'_{v,:}, \mathbf{A}'_{v,:} | y'_v = y^*, \mathbf{X}, \mathbf{A}, y] = \mathbb{P}[\mathbf{X}'_{v,:} | y'_v = y^*] \cdot \mathbb{P}[\mathbf{A}'_{v,:} | y'_v = y^*, y] \tag{25}$$

$$= \mathbb{P}[\mathbf{X}'_{v,:} | y'_v = y^*] \cdot \prod_{i \in [n] \setminus \{v\}} \mathbb{P}[\mathbf{A}'_{v,i} | y'_v = y^*, y_i] \tag{26}$$

and therefore

$$\Delta(\mathbf{A}') = \log \frac{\mathbb{P}[\mathbf{X}'_{v,:} | y'_v = y^*]}{\mathbb{P}[\mathbf{X}'_{v,:} | y'_v = 1 - y^*]} + \sum_{i \in [n] \setminus \{v\}} \underbrace{\left(\log \mathbb{P}[\mathbf{A}'_{v,i} | y'_v = y^*, y_i] - \log \mathbb{P}[\mathbf{A}'_{v,i} | y'_v = 1 - y^*, y_i]\right)}_{\Delta_i(\mathbf{A}')} \tag{27}$$

Now, to achieve (20), we want to find those structure perturbations, which lead to $\Delta(\tilde{\mathbf{A}}') < 0$ the fastest (i.e., with least changes). First, note that the first term in Equation 27 does not depend on the adjacency matrix and hence, can be ignored. The second term shows that the change in $\Delta(\mathbf{A}')$ induced by adding or removing an edges (v, i) is additive and independent of adding or removing another edge (v, j). Denote by $\tilde{\mathbf{A}}'(u)$ the adjacency matrix constructed by removing (adding) edge (v, u) from $\mathbf{A}'$ if (v, u) is (not) already in the graph. We define the change potential of node as u as $\tilde{\Delta}_u := \Delta_u(\tilde{\mathbf{A}}'(u)) - \Delta_u(\mathbf{A}')$. Then, we only need find those nodes u with maximal change potential $|\tilde{\Delta}_u| = |\Delta_u(\tilde{\mathbf{A}}'(u)) - \Delta_u(\mathbf{A}')|$ and $\tilde{\Delta}_u < 0$ and disconnect (connect) them in decreasing order of $|\tilde{\Delta}_u|$ until $\Delta(\tilde{\mathbf{A}}') < 0$. We will now show that any node u has maximal negative change potential, who either satisfies i) $y_u = y_*$ and $\mathbf{A}'_{v,u} = 1$ or ii) $y_u \neq y_*$ and $\mathbf{A}'_{v,u} = 0$.

To prove this, we make a case distinction on the existence of (v, u) in the unperturbed graph and the class of y_u:

Case $\mathbf{A}'_{v,u} = 0$:

We distinguish two subcases:

i) $y_u \neq y_*$:

We can write

$$\begin{aligned}\tilde{\Delta}_u &= \log \mathbb{P}[\mathbf{A}'_{v,i} = 1 | y'_v = y^*, y_i] - \log \mathbb{P}[\mathbf{A}'_{v,i} = 1 | y'_v = 1 - y^*, y_i] \\ &\quad - \log \mathbb{P}[\mathbf{A}'_{v,i} = 0 | y'_v = y^*, y_i] + \log \mathbb{P}[\mathbf{A}'_{v,i} = 0 | y'_v = 1 - y^*, y_i] \\ &= \log q - \log p - \log(1 - q) + \log(1 - p) \qquad (28) \\ &< 0 \qquad (29)\end{aligned}$$

Equation 28 follows from the sampling process of the CSBM. Equation 29 follows from $q < p$, implying $\log q - \log p < 0$ and $-\log(1 - q) + \log(1 - p) < 0$.

ii) $y_u = y_*$

We can write

$$\tilde{\Delta}_u = \log p - \log q - \log(1 - p) + \log(1 - q) > 0 \tag{30}$$

where the last $>$ follows similarly from $q < p$.

Case $\mathbf{A}'_{v,u} = 1$:

i) $y_u \neq y_*$:

We can write

$$\begin{aligned}\tilde{\Delta}_u = &\log \mathbb{P}[\mathbf{A}'_{v,i} = 0 | y'_v = y^*, y_i] - \log \mathbb{P}[\mathbf{A}'_{v,i} = 0 | y'_v = 1 - y^*, y_i] \\ &- \log \mathbb{P}[\mathbf{A}'_{v,i} = 1 | y'_v = y^*, y_i] + \log \mathbb{P}[\mathbf{A}'_{v,i} = 1 | y'_v = 1 - y^*, y_i] \\ &> 0 \end{aligned} \tag{31}$$

where Equation 31 follows by the insight, that $\tilde{\Delta}_u$ is the same as for case $\mathbf{A}'_{v,u} = 0$ except multiplied with -1.

ii) $y_u = y_*$

We can write

$$\tilde{\Delta}_u = \log q - \log p - \log(1-q) + \log(1-p) < 0 \tag{32}$$

where the first equality follows from the insight, that $\tilde{\Delta}_u$ is again the same as for case $\mathbf{A}'_{v,u} = 0$ except multiplied with -1. The last $>$ follows again from $q < p$.

The theorem follows from the fact that only the cases where we add an edge to a node of different class, or drop an edge to a node with the same class have negative change potential and the fact, that both cases have the same change potential.

□

C Robustness Metrics

We restate the degree corrected robustness of a classifier f w.r.t. a reference classifier g:

$$R(f, g) = \frac{1}{|V'|} \sum_{v \in V'} \frac{\text{Robustness}(f, g, v)}{\deg(v)} \tag{33}$$

Using the true labels y instead of a reference classifier g in (33), one can measure the maximal achievable robustness (before semantic content changes) as $R(g) := R(g, y)$, i.e. the semantic boundary. Note that we can exactly compute $R(g)$ due to knowledge of the data generating process. We measure **adversarial robustness** as the fraction of optimal robustness $R(g)$ achieved: $R^{adv} = R(f, g)/R(g)$. Again, to correctly measure R^{adv}, the identical attack is performed to measure $R(f, g)$ and $R(g)$.

A model f can have high adversarial- but also high over-robustness (see Section 3). To have a metric, which truly shows a complete picture of the robustness properties of a model, we take the harmonic mean of R^{adv} and the percentage of how much robustness is legitimate ($1 - R^{over}$) and define an **$\mathbf{F}_1$-robustness score**: $F_1^{rob}(\cdot, \cdot) = 2\frac{(1-R^{over}) \cdot R^{adv}}{(1-R^{over}) + R^{adv}}$. Only a model showing perfect adversarial robustness and no over-robustness achieves $F_1^{rob} = 1$.

D Experiment Details

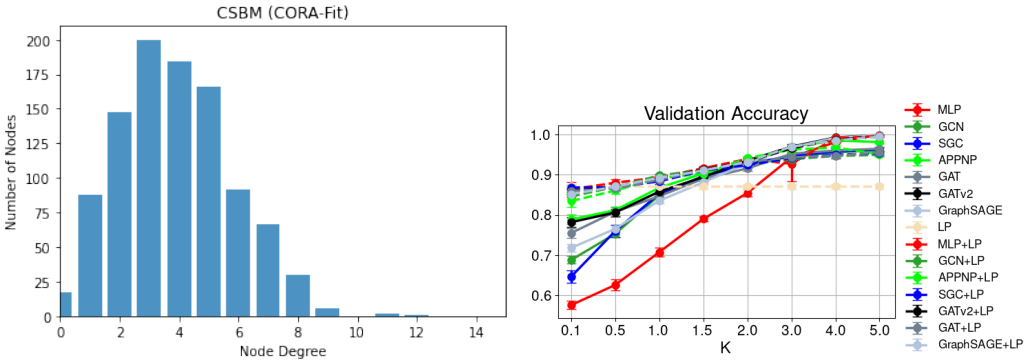
Dataset.

We use Contextual Stochastic Block Models (CSBMs). The main setup is described in Section 5. Table 3 summarizes some dataset statistics and contrasts them with CORA. The reported values are independent of K, hence we report the average across 10 sampled CSBM graphs for one K.

Figure 6a shows that CSBM graphs mainly contains low-degree nodes.

Dataset	# Nodes	# Edges	# Features	# Classes	Average Node Degree	Average Same-Class Node Degree	Average Different-Class Node Degree
CSBM	1,000	3,928±50	21	2	3.92±0.05	3.25±0.05	0.67±0.02
CORA	2,708	10,556	1,433	7	3.90	3.16	0.74

Table 3: Dataset statistics



(a) (b)

Figure 6: (a) Degree distribution of a CSBM graph as parametrized in Section 5 ($n = 1000$). (b) Validation accuracies of the best performing hyperparameters of the different models. Note that GNNs for low K (high-structure relevance) underperform pure LP.

Graph Neural Networks and Label Propagation (LP).

We perform extensive hyperparameter search for LP and MLP and each GNN model for each individual K and choose, for each K, the on average best performing hyperparamters on 10 graphs sampled from the respective CSBM. For the *MODEL+LP* variants of our models, we use the individually best performing hyperparameters of the model and LP. Interestingly, we find that very different hyperparameters are optimal for different choices of the feature-information defining parameter K. We also find that using the default parameters from the respective model papers successful on the benchmark real-world datasets, don't work well for some choices of K, especially low K when structure is very important but features not so.

We train all model for 3000 epochs with a patients of 300 epochs using Adam (Kingma & Ba, 2015) and explore learning rates $[0.1, 0.01, 0.001]$ and weight decay $[0.01, 0.001, 0.001]$ and additionally for

- *MLP:* We use a 1 (Hidden)-Layer MLP and test hidden dimensions $[32, 64, 128, 256]$ and dropout $[0.0, 0.3, 0.5]$. We employ the ReLU activation function.
- *LP:* We use 50 iterations and test α in the range between 0.00 and 1.00 in step sizes of 0.05. LP is the only method having the same hyperparameters on all CSBMs, as it is independent of K.
- *SGC:* We explore $[1, 2, 3, 4, 5]$ number of hops and additionally, a learning rate of 0.2. We investigate dropouts of $[0, 0.3, 0.5]$. SGC was the most challening to train for low K.
- *GCN:* We use a two layer (ReLU) GCN with 64 filters and dropout $[0.0, 0.3, 0.5]$
- *GAT:* We use a two layer GAT with 8 heads and 8 features per head with LeakyReLU having a negative slope of 0.2. We test dropout $[0.0, 0.3, 0.6]$ and neighbourhood dropout $[0.0, 0.3, 0.6]$.
- *GATv2:* We use the best performing hyperparameters of GAT.
- *APPNP:* We use 64 hidden layers, $K = 10$ iterations, dropout $[0.0, 0.3, 0.6]$ and $[0.0, 0.3, 0.5]$ and test α in $[0.05, 0.1, 0.2]$. Interestingly, the higher K, we observe higher α performing better.

Further details, such as the best performing hyperparameters, can be found in the released experiment configuration files and source code. The (averaged) validation accuracies can be seen in Figure 6b.

Attacks.

- *Nettack* attacks a surrogate SGC model to approximately find maximal adversarial edges. As a surrogate model, instead of using a direct SGC implementation as in the models section, we use a 2-Layer GCN with the identity function as non-linearity and 64 filters and found it trains easier on $K = 0.1$ and hence, provides better adversarial examples for $K = 01$ than direct SGC implementation. For higher K, differences are neglectable. We use the same hyperparamter search as outlined for the conventional GCN.
- *DICE* randomly disconnects d edges from the test node v to same-class nodes and connects b edges from v to different-class nodes. For a given local budget Δ, we set $d = 0$ and $b = \Delta$.

Further details, such as the best performing hyperparameters, can be found in the released experiment configuration files and source code.

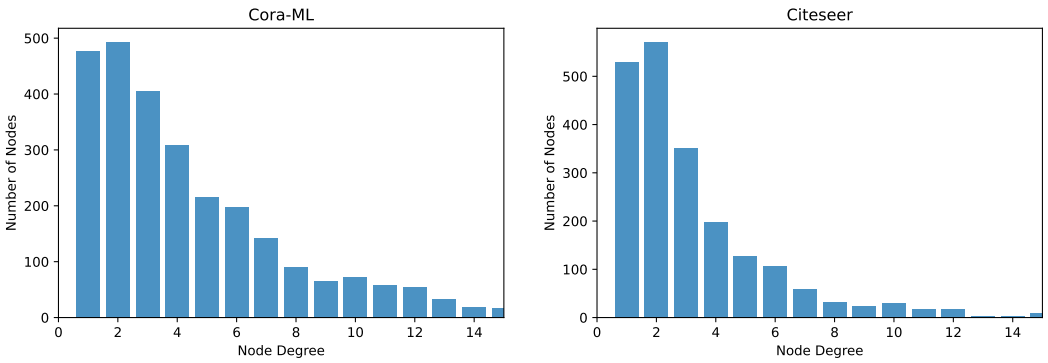
D.1 Real-World Graphs

We introducing the experimental setup for evaluating overrobustness of GNNs on real-world graphs in detail. This includes datasets, models and evaluation procedure.

Datasets To explore overrobustness on real-world graphs, the citation networks Cora-ML (Bojchevski & Günnemann, 2017) and Citeseer (Sen et al., 2008) are selected. Table 4 provides an overview over the most important dataset characteristics. Figure 7 visualizes the degree distributions up to degree 15. For both datasets, 40 nodes per class are randomly selected as validation and test nodes. The remaining nodes are selected as labeled training set. On Cora-ML, this results in approximately 80% training, 10% validation and 10% test nodes. Following the inductive approach used in Section 5 for CSBMs, model optimization is performed using the subgraph spanned by all training nodes. Early stopping uses the subgraph spanned by all training and validation nodes.

Dataset	# Nodes	# Edges	# Features	# Classes	Average Node Degree	Average Same-Class Node Degree	Average Different-Class Node Degree
Cora-ML	2,810	15,962	2,879	7	5.68	4.46	1.22
Citeseer	2,110	7,336	3,703	6	3.48	2.56	0.92

Table 4: Dataset statistics



(a) Cora-ML (b) Citeseer

Figure 7: Degree distribution by dataset.

Model Architectures We evaluate the robustness of Graph Convolutional Networks (GCN), Label Propagation (LP), and GCN followed by LP post-processing (GCN+LP). The GCN architecture

and optimization scheme follow Geisler et al. (2021). The GCN has two layers 64 filters. During training, a dropout of 0.5 is applied. We optimize the model parameters for a maximum of 3000 epochs using Adam (Kingma & Ba, 2015) with learning rate 0.01 and weight decay 0.001. LP uses the normalized adjacency as transition matrix and is always performed for ten iterations. This mirrors related architectures like APPNP (Gasteiger et al., 2019). We additionally choose $\alpha = 0.7$ on both datasets by grid-search over $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For GCN+LP, it should be noted that LP is applied as a post-processing routine at test-time only. It is not included during training.

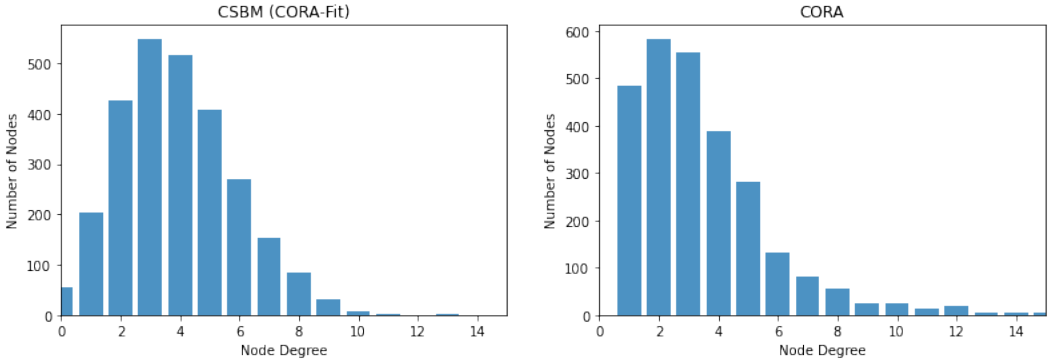
Evaluating Degree-Depending Robustness

We investigate degree-dependent robustness of GNNs by following a similar strategy as the ℓ_2-weak attack on CSBMs. However, real-world datasets are multi-class. Therefore, we ensure to only connect the target nodes to nodes of one selected class. More concretely, let v be a correctly classified test node with label c^*. We investigate for each class $c \neq c^*$, how many edges we can connect to v, until the model's prediction changes and denote this number $N_c(v)$. The attack then works as follows: First, we project the high-dimesional feature vectors into lower dimensional space by applying the first weight matrix of the GCN. Then, we iteratively add edges connecting v to the most similar nodes (after projection) in ℓ_2-norm from c and evaluate after how many insertions the model's prediction changes. We present the results for the class achieving lowest robustness, i.e. smallest $N_c(v)$, and the results for the class achieving highest robustness, i.e. largest $N_c(v)$.

E Further Results

E.1 Degree Distribution CSBM vs CORA

Note that degrees in CSBM do **not** follow a power-law distribution. However, they are similar in a different sense to common benchmark citation networks. The goal of this section is to show that the large majority of nodes in both graphs have degrees 2, 3, 4 or 5. Low degree nodes are even more pronounced in CORA than CSBMs.



(a) Graph sampled from a CSBM, using $n = 2708$ as CORA. Note that the graph structure is of a CSBM is independent of the K but only dependent on p, q which have been set to fit CORA. Plot cut at node degree 15.

(b) CORA contains mainly low degree nodes. Plot cut at node degree 15.

Figure 8: Degree distribution of the used CSBMs vs CORA, both distribution show that the graphs mainly contain low-degree nodes. This is even more pronounced in CORA than CSBMs.

E.2 Test-Accuracy on CSBM

This section summaries the detailed performance of the different models on the CSBMs.

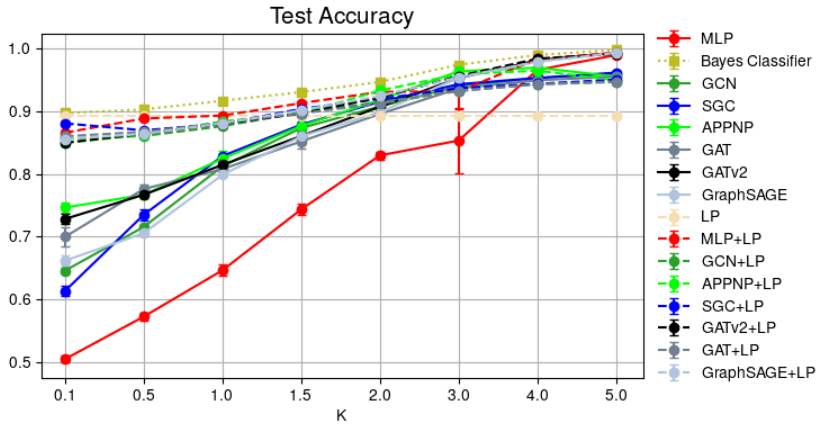


Figure 9: Test accuracy of models on test nodes on the CSBMs.

	0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
Bayes Classifier (BC)	**89.7%**	**90.3%**	**91.7%**	**93.1%**	**94.7%**	**97.4%**	**99.0%**	**99.8%**
BC (Features Only)	50.8%	59.0%	68.4%	76.5%	83.4%	92.6%	97.5%	99.3%
BC (Structure Only)	89.8%	89.8%	89.8%	89.8%	89.8%	89.8%	89.8%	89.8%
MLP	50.4%	57.2%	64.6%	74.3%	83.0%	85.3%	96.6%	99.0%
GCN	64.6%	71.5%	81.2%	87.3%	90.8%	94.0%	95.3%	96.0%
SGC	61.3%	73.5%	82.9%	87.9%	91.5%	94.3%	95.3%	96.2%
APPNP	74.6%	76.7%	82.4%	87.7%	91.8%	**96.4%**	97.0%	95.5%
GAT	70.0%	77.6%	80.8%	85.2%	89.6%	93.7%	95.0%	95.5%
GATv2	72.8%	76.7%	81.5%	86.1%	90.7%	95.3%	98.1%	99.3%
GraphSAGE	66.2%	70.6%	79.9%	86.0%	89.8%	95.4%	97.9%	**99.4%**
LP	**89.2%**	**89.2%**	89.2%	89.2%	89.2%	89.2%	89.2%	89.2%
MLP+LP	86.6%	88.8%	**89.3%**	**91.3%**	93.1%	93.2%	**98.4%**	99.3%
GCN+LP	85.6%	86.1%	87.6%	89.8%	91.5%	93.2%	94.3%	94.7%
APPNP+LP	85.0%	86.3%	88.4%	89.6%	**93.3%**	95.9%	96.5%	95.0%
SGC+LP	88.1%	87.0%	88.0%	90.4%	92.0%	93.7%	94.4%	95.0%
GATv2+LP	85.0%	86.4%	88.0%	89.6%	92.2%	95.6%	**98.4%**	99.2%
GAT+LP	86.0%	86.7%	88.2%	89.7%	91.2%	93.3%	94.3%	94.7%
GraphSAGE+LP	85.5%	86.4%	88.1%	90.2%	92.4%	95.4%	97.9%	99.3%

Table 5: Average test accuracies of the models on the sampled test nodes on the CSBMs.

	0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
MLP	0.5%	0.6%	0.9%	0.9%	0.6%	5.2%	0.3%	0.1%
Bayes Classifier (BC)	0.2%	0.2%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%
BC (Features Only)	1.5%	1.5%	1.7%	1.8%	1.6%	1.1%	0.6%	0.3%
BC (Structure Only)	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%	0.8%
GCN	0.5%	0.5%	0.5%	0.3%	0.2%	0.2%	0.2%	0.1%
SGC	0.8%	0.8%	0.4%	0.2%	0.1%	0.2%	0.1%	0.1%
APPNP	0.9%	0.8%	1.2%	0.7%	0.3%	0.2%	0.7%	0.9%
GAT	1.5%	0.6%	1.1%	1.1%	0.4%	0.3%	0.3%	0.4%
GATv2	0.8%	0.4%	0.4%	0.5%	0.3%	0.5%	0.2%	0.1%
GraphSAGE	0.8%	0.5%	0.3%	0.3%	0.2%	0.1%	0.2%	0.1%
LP	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
MLP+LP	1.7%	0.2%	0.3%	0.3%	0.3%	2.8%	0.2%	0.1%
GCN+LP	0.3%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%
APPNP+LP	0.4%	0.2%	0.3%	0.4%	0.2%	0.2%	0.5%	0.5%
SGC+LP	0.3%	0.3%	0.3%	0.1%	0.2%	0.1%	0.2%	0.2%
GATv2+LP	0.3%	0.4%	0.2%	0.5%	0.2%	0.2%	0.2%	0.2%
GAT+LP	0.3%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%
GraphSAGE+LP	0.5%	0.3%	0.2%	0.2%	0.2%	0.2%	0.1%	0.1%

Table 6: Standard deviation of test accuracy of the models on the sampled test nodes on the CSBMs.

E.3 EXTENT OF SEMANTIC CONTENT CHANGE IN COMMON PERTURBATION MODELS

The extent of semantic content change looks similar for DICE and Nettack to Table 2 (ℓ_2-weak).

Nettack:

Furthermore, for $K = 0.1$ the SGC uses by Nettack has only mediocre test-accuracy due to features not being very informative. Therefore, Nettack sometimes proposes to add same-class edges or remove different-class edges.

Threat Models	K=0.1	K=0.5	K=1.0	K=1.5	K=2.0	K=3.0	K=4.0	K=5.0
$\mathcal{B}_1(\cdot)$	10.6	9.7	9.1	6.8	4.4	1.9	0.7	0.2
$\mathcal{B}_2(\cdot)$	23.5	24.7	24.8	19.8	14.1	6.2	2.2	0.7
$\mathcal{B}_3(\cdot)$	35.7	41.0	43.6	38.1	28.8	14.2	4.9	1.6
$\mathcal{B}_4(\cdot)$	47.3	55.0	61.2	57.5	46.8	25.1	9.2	3.2
$\mathcal{B}_{\text{deg}}(\cdot)$	45.4	42.9	50.0	47.6	39.4	21.9	8.9	3.1
$\mathcal{B}_{\text{deg+2}}(\cdot)$	63.9	73.7	90.1	89.8	79.6	50.2	23.6	8.6

Table 7: Percentage (%) of nodes for which we find perturbed graphs in $\mathcal{B}_\Delta(\cdot)$ violating semantic content preservation, i.e. with changed ground truth labels. Calculated by adding or dropping Δ edges suggested by Nettack to every target node. Note that for K=4.0 and K=5.0 structure is not necessary for good generalization (Table 1). Standard deviation are insignificant and hence, omitted.

DICE:

Threat Models	K=0.1	K=0.5	K=1.0	K=1.5	K=2.0	K=3.0	K=4.0	K=5.0
$\mathcal{B}_1(\cdot)$	14.9	10.9	9.0	6.2	4.3	2.1	0.6	0.2
$\mathcal{B}_2(\cdot)$	36.8	31.4	26.2	19.7	13.8	6.4	2.0	0.6
$\mathcal{B}_3(\cdot)$	58.6	52.9	46.4	37.7	28.8	13.9	5.0	1.3
$\mathcal{B}_4(\cdot)$	77.1	72.6	66.6	57.0	45.8	25.8	9.6	2.9
$\mathcal{B}_{\text{deg}}(\cdot)$	76.6	58.9	55.6	48.9	38.5	22.1	8.7	2.8
$\mathcal{B}_{\text{deg+2}}(\cdot)$	100.0	100.0	99.2	92.2	79.9	50.8	24.0	8.2

Table 8: Percentage (%) of nodes for which we find perturbed graphs in $\mathcal{B}_\Delta(\cdot)$ violating semantic content preservation, i.e. with changed ground truth labels. Calculated by randomly connecting Δ different-class nodes (DICE) to every target node. Note that for K=4.0 and K=5.0 structure is not necessary for good generalization (Table 1). Standard deviation are insignificant and hence, omitted.

E.4 OVER-ROBUSTNESS OF GRAPH NEURAL NETWORKS

ℓ_2-weak:

K	0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
MLP	43.1%	35.6%	30.7%	25.5%	19.3%	9.9%	3.8%	1.3%
GCN	35.3%	30.3%	25.7%	19.8%	14.1%	5.2%	1.6%	0.4%
SGC	37.4%	29.6%	23.9%	20.1%	14.1%	5.2%	2.3%	0.6%
APPNP	32.5%	30.1%	24.3%	19.6%	16.8%	7.0%	2.4%	0.9%
GAT	33.3%	29.2%	26.6%	21.2%	15.7%	7.3%	2.9%	0.9%
GATv2	33.0%	29.4%	26.6%	21.2%	16.1%	7.5%	2.9%	0.9%
GraphSAGE	36.7%	32.3%	26.8%	22.2%	16.9%	8.3%	3.3%	1.2%
LP	16.0%	13.7%	11.3%	8.9%	6.7%	3.0%	1.0%	0.3%
MLP+LP	21.4%	14.8%	15.2%	14.4%	13.1%	6.6%	2.5%	0.9%
GCN+LP	23.5%	20.9%	19.3%	15.2%	10.3%	3.7%	1.2%	0.3%
APPNP+LP	24.5%	22.0%	18.2%	14.8%	11.7%	4.4%	1.6%	0.5%
SGC+LP	19.1%	18.2%	17.3%	14.8%	9.9%	3.8%	1.5%	0.4%
GATv2+LP	25.1%	20.8%	18.2%	15.7%	11.1%	5.1%	2.2%	0.6%
GAT+LP	21.5%	20.5%	17.3%	14.4%	10.8%	4.9%	1.9%	0.6%
GraphSAGE+LP	23.9%	21.2%	18.6%	15.8%	12.5%	5.6%	2.3%	0.9%

Table 9: Over-Robustness R^{over} measured using ℓ_2-weak (see also Figure 3a with a budget of the degree of the target node. Standard deviations never exceed 1% except for MLP+LP at $K = 0.1$ which has a standard deviation of 3%.

Nettack:

Table 10 shows that over-robustness is not only occurring for weak attacks. Especially, the MLP results show that if we would have a classifier perfectly robust against Nettack in the bounded perturbation set, for all $K \leq 2$ (where structure matters), this would result in high over-robustness.

	0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
MLP	26.5%	26.4%	28.4%	25.0%	19.2%	9.9%	3.8%	1.3%
GCN	13.5%	11.4%	6.0%	2.4%	0.8%	0.1%	0.0%	0.0%
SGC	7.7%	4.6%	2.9%	1.4%	0.4%	0.1%	0.1%	0.0%
APPNP	11.9%	10.1%	9.0%	3.8%	1.4%	0.1%	0.6%	0.6%
GAT	9.7%	15.1%	12.7%	11.0%	5.4%	3.3%	1.6%	0.5%
GATv2	14.0%	12.4%	13.3%	13.8%	8.8%	5.1%	2.7%	0.8%
GraphSAGE	14.6%	17.7%	15.7%	12.3%	9.9%	4.8%	2.1%	0.7%
LP	5.4%	5.4%	5.6%	5.4%	4.2%	1.9%	0.6%	0.2%
MLP+LP	9.4%	6.9%	10.2%	9.6%	10.1%	5.0%	1.7%	0.6%
GCN+LP	4.5%	3.6%	2.0%	1.1%	0.5%	0.2%	0.1%	0.1%
APPNP+LP	8.5%	5.6%	3.1%	2.1%	0.3%	0.1%	0.4%	0.3%
SGC+LP	1.0%	1.1%	1.1%	0.5%	0.2%	0.1%	0.1%	0.0%
GATv2+LP	8.9%	6.3%	6.3%	5.2%	5.3%	3.1%	1.7%	0.5%
GAT+LP	6.5%	6.8%	4.4%	3.7%	2.3%	1.3%	0.7%	0.3%
GraphSAGE+LP	7.7%	7.4%	7.1%	4.8%	4.1%	1.7%	0.8%	0.4%

Table 10: Over-Robustness R^{over} measured using Nettack with a budget of the degree of the target node. Standard deviations rarely exceed 1% notably for MLP at $K = 0.1$ with 4.8% and MLP+LP at $K = 0.1$ at 3%.

DICE:

Table 11 shows, as for Nettack, that over-robustness is not only occurring for weak attacks. Especially, the MLP results show that if we would have a classifier perfectly robust against DICE in the bounded perturbation set, for all $K \leq 2$ (where structure matters), this would result in high over-robustness.

	0.1	0.5	1.0	1.5	2.0	3.0	4.0	5.0
MLP	43.6%	35.0%	30.8%	25.0%	18.8%	10.0%	3.5%	1.1%
GCN	34.0%	29.2%	23.2%	16.8%	11.1%	3.4%	0.9%	0.3%
SGC	37.2%	28.9%	21.3%	16.5%	10.7%	3.3%	0.9%	0.2%
APPNP	31.1%	27.8%	22.7%	17.2%	13.6%	4.3%	1.7%	0.8%
GAT	32.9%	27.9%	25.7%	19.2%	13.1%	5.7%	2.2%	0.7%
GATv2	31.2%	27.8%	23.6%	18.7%	13.5%	6.6%	2.4%	0.8%
GraphSAGE	34.7%	29.9%	24.7%	19.4%	14.6%	7.0%	2.6%	0.9%
LP	16.0%	13.4%	10.8%	8.7%	6.4%	2.7%	0.9%	0.3%
MLP+LP	20.8%	14.7%	15.3%	14.5%	12.9%	6.9%	2.4%	0.8%
GCN+LP	21.9%	20.0%	17.0%	12.5%	8.2%	2.5%	0.8%	0.3%
APPNP+LP	23.4%	20.1%	15.9%	13.2%	8.7%	2.7%	1.1%	0.5%
SGC+LP	18.5%	17.6%	15.3%	11.9%	7.6%	2.6%	0.8%	0.2%
GATv2+LP	22.7%	19.5%	15.7%	13.3%	8.7%	4.6%	1.8%	0.6%
GAT+LP	21.9%	18.9%	15.6%	12.3%	8.9%	3.5%	1.5%	0.5%
GraphSAGE+LP	22.8%	19.2%	16.0%	12.4%	9.5%	4.2%	1.7%	0.7%

Table 11: Over-Robustness R^{over} measured using Nettack with a budget of the degree of the target node. Standard deviations are insignificant and removed for brevity.

E.5 Adversarial-Robustness of Graph Neural Networks

E.5.1 Nettack

Nettack is the strongest attack we employ, hence, it is the best heuristic we have to measure (semantic-aware) adversarial robustness R^{adv} (see Metric-Section C) and conventional, non-semantics aware, adversarial robustness $R(f)$ (see Section 5). Figure 10 shows that surprisingly, MLP+LP has highest adversarial robustness, if structure matters $K \leq 3$.

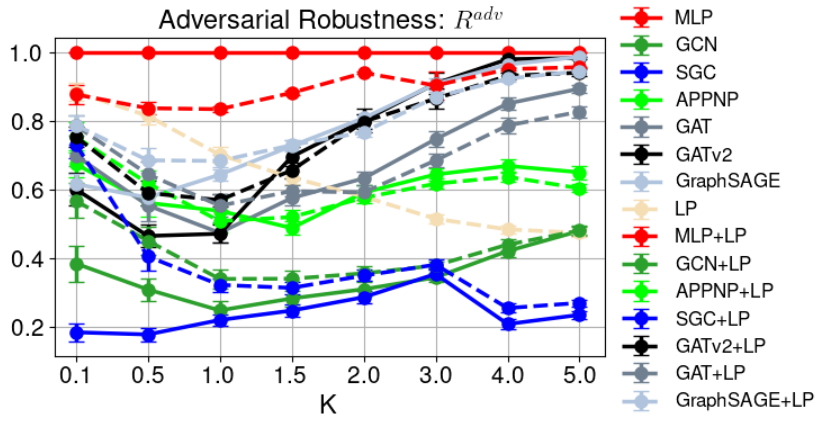


Figure 10: Semantic Aware Robustness R^{adv} measured using Nettack. Dashed lines are LP models.

Figure 11 shows that similar, albeit slightly less accurate adversarial robustness measurements are obtained by not including the the semantic awareness. Rankings can indeed change, but only if these models are already very close regarding R^{adv}.

The harmonic mean of R^{adv} and R^{over} (see Metric-Section C) shows a complete picture of the robustness of the analysed models. To measure R^{over}, we use ℓ_2-weak. Note that no GNN achieves top placements using this ranking, but the best models, depending on the amount of feature information, are LP, MLP+LP and MLP.

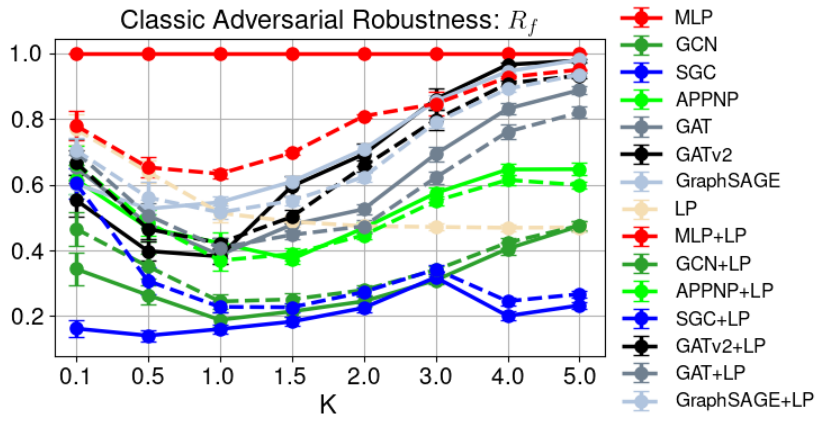


Figure 11: Conventional (Degree-Corrected) Robustness R^{adv} measured using Nettack. Dashed lines are LP models.

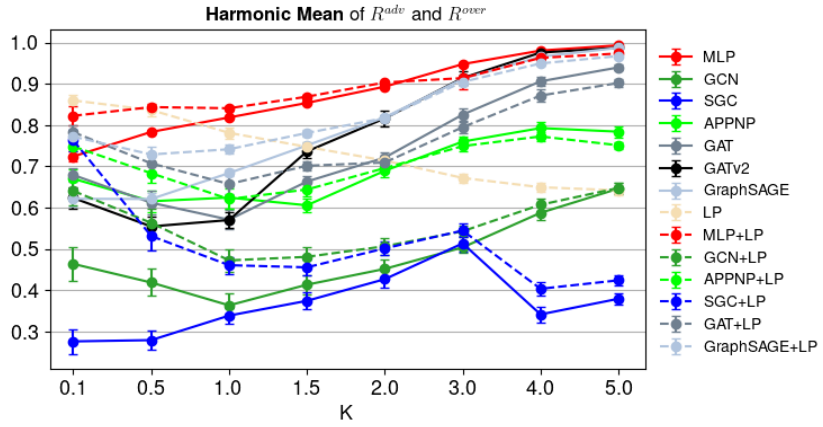


Figure 12: The harmonic mean of R^{adv} and R^{over}, with R^{adv} measured using Nettack and R^{over} using ℓ_2-weak.

E.5.2 DICE

DICE, as just randomly connecting to different class nodes, turns out to be a very weak attack similar to ℓ_2-weak. Hence, its adversarial robustness counts differ significantly from the stronger Nettack. We find that with DICE we measure significantly higher over-robustness (Section E.4). Here, we also note a significant difference between true (semantic aware) adversarial robustness as presented in 13, where MLP+LP is best with LP models generally performing on par with GNNs. Figure 14 shows that using conventional adversarial robustness, when having a weak attack, results in a significantly different picture to the true adversarial robustness.

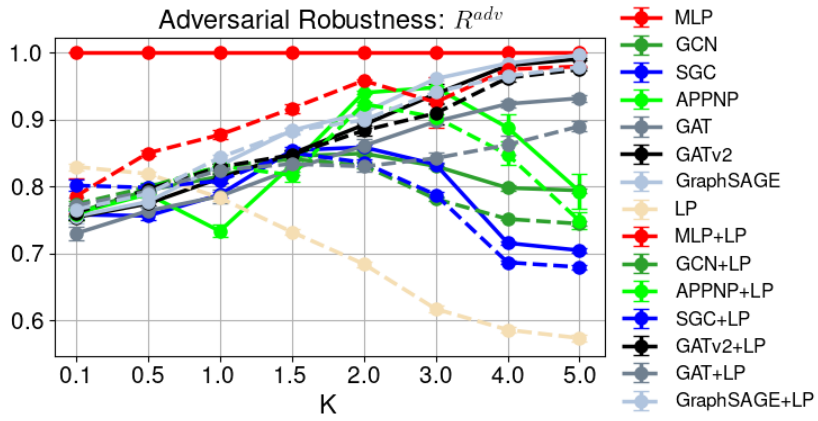


Figure 13: Semantic Aware Robustness R^{adv} measured using DICE. Dashed lines are LP models.

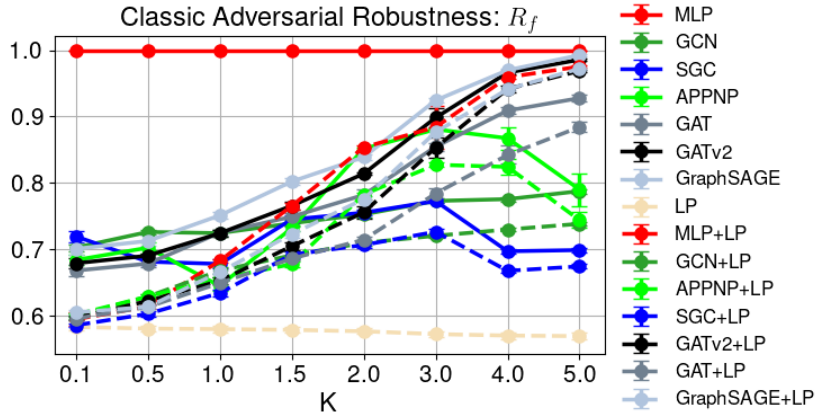


Figure 14: Conventional (Degree-Corrected) Robustness R^{adv} measured using DICE. Dashed lines are LP models.

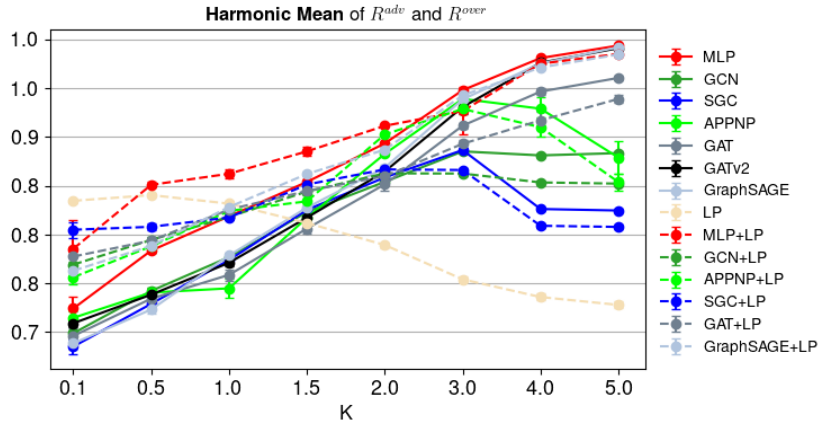


Figure 15: The harmonic mean of R^{adv} and R^{over}, with R^{adv} measured using DICE and R^{over} using ℓ_2-weak.

E.5.3 STRUCTURE PERTURBATIONS UNTIL GCN-PREDICTION CHANGES ON CSBMS

Figure 16 shows that on a CSBM with $K = 1.5$, where a GCN already shows significant over-robustness (see Figure 3b), the GCN is less strongly robustness as on Cora-ML (compare to Figure 4).

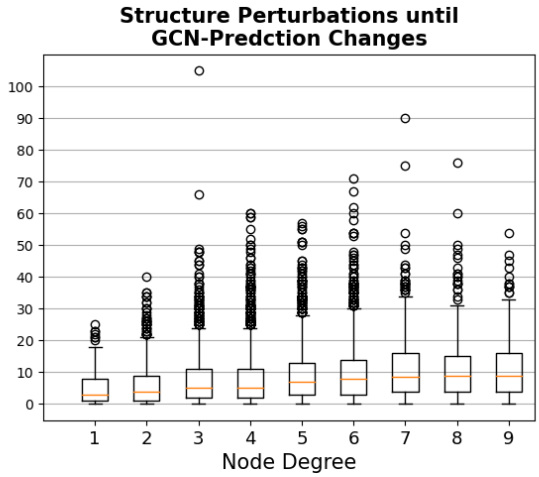


Figure 16: Robustness of GCN predictions on CSBMs with $K = 1.5$ as shown in Figure 1.

E.5.4 AVERAGE ROBUSTNESS BAYES CLASSIFIER

Figure 16 shows that the average robustness of the Bayes classifier increases with K and is linear in the degree of the node. Furthermore, if structure matters ($K \leq 3$), average robustness rarely exceeds the degree of a node.

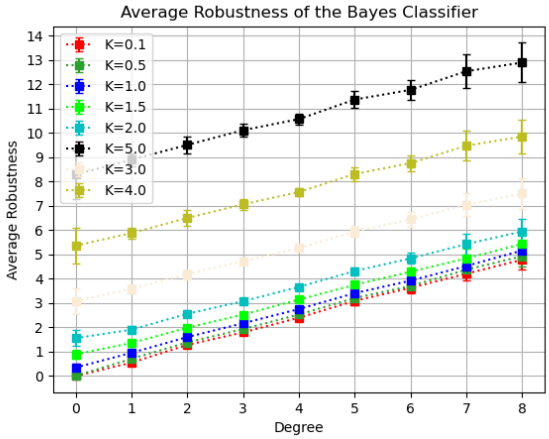
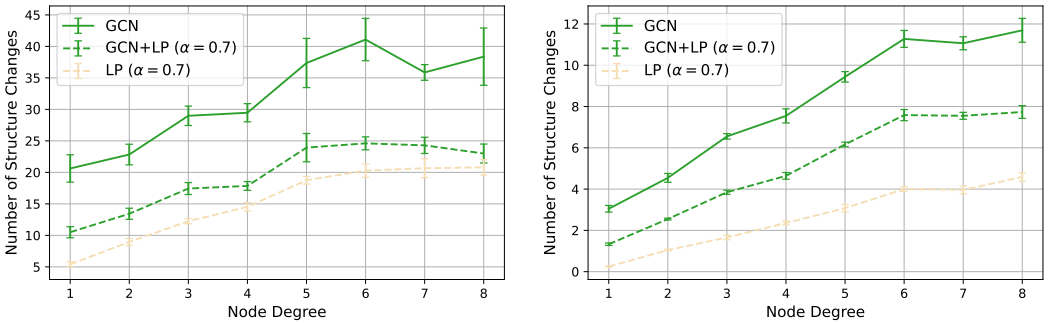


Figure 17: Average Robustness of the Bayes classifier on CSBMs.

E.5.5 FURTHER RESULTS ON REAL-WORLD GRAPHS

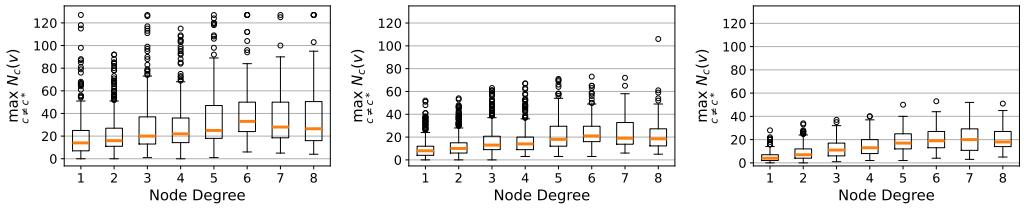
Model **Dataset**	GCN	GCN+LP	LP
Citeseer	69.4 ± 1.75	69.5 ± 1.62	66.5 ± 2.12
Cora-ML	87.1 ± 2.12	86.7 ± 1.80	84.0 ± 2.45

Table 12: Model test accuracy and standard deviation over eight data splits



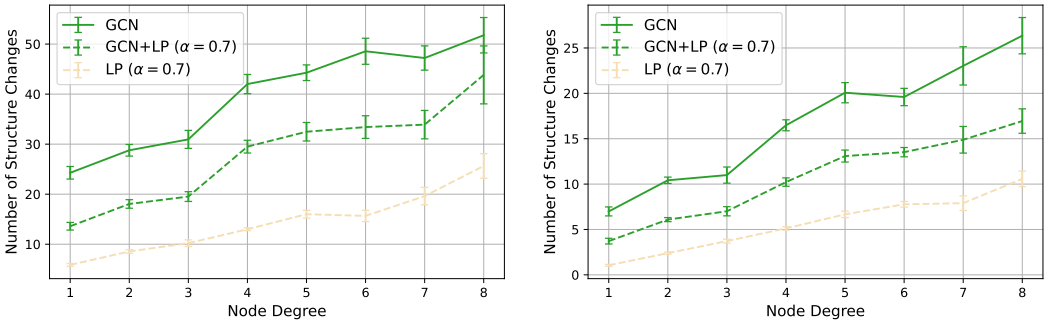
(a) Maximal (per-class) robustness (b) Minimal (per-class) robustness

Figure 18: Mean robustness per node degree on the Cora-ML dataset. Error bars indicate the standard error of the mean over eight data splits.



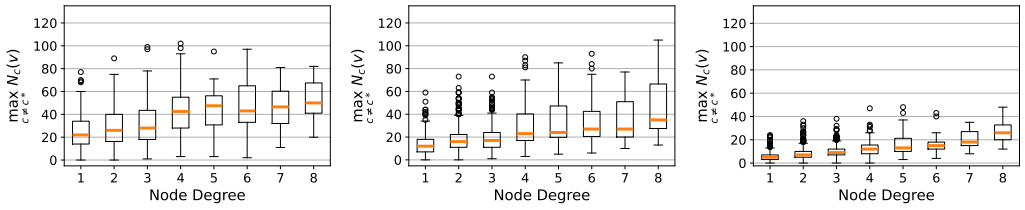
(a) GCN (b) GCN+LP (c) LP

Figure 19: Distribution of maximal (per-class) node robustness by node degree on the Cora-ML dataset for different models. Results are aggregated over eight data splits.



(a) Maximal (per-class) robustness (b) Minimal (per-class) robustness

Figure 20: Mean robustness per node degree on the Citeseer dataset. Error bars indicate the standard error of the mean over eight data splits.



(a) GCN (b) GCN+LP (c) LP

Figure 21: Distribution of maximal (per-class) node robustness by node degree on the Citeseer dataset for different models. Results are aggregated over eight data splits.

F ADDITIONAL RELATED WORK

We see us related to works using synthetic graph models to generate principled insights into GNNs. Notably, Fountoulakis et al. (2022) show that in non-trivially CSBMs settings (hard regime), GATs, with high probability, can't distinguish same-class edges from different-class edges and degenerate to GCNs. Baranwal et al. (2021) study GCNs on CSBMs and find graph convolutions extent the linear separability of the data. Palowitch et al. (2022) generate millions of synthetic graphs to explore the performance of common GNNs on graph datasets with different characteristics to the common benchmark real-world graphs. However, their studied degree-corrected SBM is fundamentally limited to transductive learning.

Regarding the bigger picture in robust graph learning, all works measuring small changes to the graph's structure using the ℓ_0-norm can be seen as related. This is a large body of work and includes but is not limited to i) the attack literature such as (Zügner et al., 2018; Dai et al., 2018; Waniek et al., 2018; Chen et al., 2018; Zügner & Günnemann, 2019a; Geisler et al., 2021); ii) various defenses ranging from detecting attacks (Wu et al., 2019b; Entezari et al., 2020), proposing new robust layers and architectures (Zhu et al., 2019; Geisler et al., 2020) to robust training schemes (Zügner & Günnemann, 2019b; Xu et al., 2019; 2020); iii) robust certification (Bojchevski et al., 2020; Schuchardt et al., 2021). An overview of the adversarial robustness literature on GNNs is given by Günnemann (2022). Zheng et al. (2021) provide a graph robustness benchmark.

Regarding sound perturbations models, distantly related is also the work of Geisler et al. (2022), which apply GNNs to combinatorial optimization tasks and therefore, can describe how the perturbations change or preserve the label and thereby, semantics.