How does Heterophily Impact the Robustness of Graph Neural Networks? Theoretical Connections and Practical Implications

Jiong Zhu  
University of Michigan  
jiongzhu@umich.edu

Junchen Jin  
Northwestern University  
mark.jin@u.northwestern.edu

Donald Loveland  
University of Michigan  
dlovelan@umich.edu

Michael T. Schaub  
RWTH Aachen University  
schaub@cs.rwth-aachen.de

Danai Koutra  
University of Michigan  
dkoutra@umich.edu

ABSTRACT
We bridge two research directions on graph neural networks (GNNs), by formalizing the relation between heterophily of node labels (i.e., connected nodes tend to have dissimilar labels) and the robustness of GNNs to adversarial attacks. Our theoretical and empirical analyses show that for homophilous graph data, impactful structural attacks always lead to reduced homophily, while for heterophilous graph data the change in the homophily level depends on the node degrees. These insights have practical implications for defending against attacks on real-world graphs: we deduce that separate aggregators for ego- and neighbor-embeddings, a design principle which has been identified to significantly improve prediction for heterophilous graph data, can also offer increased robustness to GNNs. Our comprehensive experiments show that GNNs merely adopting this design achieve improved empirical and certifiable robustness compared to the best-performing unvaccinated model. Additionally, combining this design with explicit defense mechanisms against adversarial attacks leads to an improved robustness with up to 18.3% performance increase under attacks compared to the best-performing vaccinated model.

CCS CONCEPTS
• Computing methodologies → Semi-supervised learning settings;  
• Neural networks; • Security and privacy;

KEYWORDS
graph neural networks, adversarial attacks, heterophily, structural perturbation, robustness, relation

1 INTRODUCTION
Graph neural networks (GNNs) aim to translate the enormous empirical success of deep learning to data defined on non-Euclidean domains, such as manifolds or graphs [6], and have become important tools to solve a variety of learning problems for graph structured and geometrically embedded data. However, recent works show that GNNs—much like their “standard” deep learning counterparts—have a high sensitivity to adversarial attacks: intentionally introduced minor changes in the graph structure can lead to significant changes in performance. This finding, first articulated by Zügner et al. [52] and Dai et al. [10], has triggered studies that investigated different attack scenarios [24, 30, 41, 42].

A different aspect of GNNs that has been scrutinized recently is that most GNNs do not perform well with many heterophilous datasets. GNNs generally perform well under homophily (or assortativity), i.e., the tendency of nodes with similar features or class labels to connect [35, 47]. Such datasets are thus called homophilous (or assortative). While homophilous datasets dominate the study of networks, homophily is not a universal principle; certain networks, such as romantic relationship networks or predator-prey networks in ecology, are mostly heterophilous (or disassortative). Employing a GNN which does not account for heterophily can lead to significant performance loss in heterophilous settings [1, 2, 47]. Previous works have thus proposed architectures for heterophilous data.

While previous work has focused on naturally-occurring heterophily, heterophilous interactions may also be introduced as adversarial noise: as many GNNs exploit homophilous correlation, they can be sensitive to changes that render the data more heterophilous. A natural follow-up question is if and how this observation manifests itself in previously proposed attacking strategies on GNNs. In this work, we thus investigate the relation between heterophily and robustness of GNNs against adversarial attacks of graph structure, focusing on semi-supervised node classification. More specifically, our main contributions are:

• Formalization: We formalize the relation between adversarial structural attacks and the change of homophily level in the underlying graphs with theoretical (§3.1) and empirical (§5.1) analysis. Specifically, we show that on homophilous graphs, effective structural attacks lead to increased heterophily, while, on heterophilous graphs, they alter the homophily level contingent on node degrees. To our knowledge, this is the first formal analysis of such kind.

• Heterophily-inspired Design: We show how the relation between attacks and heterophily can inspire more robust GNNs by
demonstrating that a key architectural feature in handling heterophily, separate aggregators for ego- and neighbor-embeddings, also improves the robustness of GNNs against attacks (§3.2).

- **Extensive Empirical Analysis:** We show the effectiveness of the heterophilous design in improving empirical (§5.2) and certifiable (§5.3) robustness of GNNs with extensive experiments on real-world homophilous and heterophilous datasets. Specifically, we compare GNNs with this design, which we refer to as heterophilically-adjusted GNNs, to non-adjusted models, including state-of-the-art models designed with robustness in mind. We find that heterophily-adjusted GNNs are up to 11.1 times more certifiably robust and have stronger performance under attacks by up to 40.0% compared to non-adjusted, standard models. Moreover, this design can be combined with existing vaccination mechanisms, yielding up to 18.3% higher accuracy under attacks than the best non-adjusted vaccinated model. Our code is available at https://github.com/GemsLab/HeteRobust.

## 2 NOTATION AND PRELIMINARIES

Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}) \) be a simple graph with node set \( \mathcal{V} \), edge set \( \mathcal{E} \) and node attributes \( \mathcal{X} \). The one-hop neighborhood \( N(v) = \{(u, v) \in \mathcal{E} : (u, v) \in \mathcal{E} \} \) of a node \( v \in \mathcal{V} \) is the set of all nodes directly adjacent to \( v \); the \( k \)-hop neighborhood of \( v \in \mathcal{V} \) is the set of nodes reachable by a shortest path of length \( k \). We represent the graph \( \mathcal{G} \) algebraically by an adjacency matrix \( A \in \{0, 1\}^{\mathcal{V} \times \mathcal{V}} \) and node feature matrix \( \mathcal{X} \in \mathbb{R}^{\mathcal{V} \times F} \). We use \( A_k = A + I \) to denote the adjacency matrix with self-loops added, and denote the corresponding row-stochastic matrices as \( \bar{A} = D^{-1}A \) and \( \bar{A}_k = D_k^{-1}A_k \), respectively, where \( D \) is a diagonal matrix with \( D_{kk} = \sum_j A_{kj} \) (\( D_k \) is defined analogously). We further assume that there exists a unique class label \( y_v \) for each node \( v \). Given a training set \( \mathcal{T}_Y = \{(x_1, y_1), (x_2, y_2), \ldots\} \) of labeled nodes, the goal of semi-supervised node classification is to learn a mapping \( f : \mathcal{V} \rightarrow \mathcal{Y} \) from the nodes to the set \( \mathcal{Y} \) of class labels.

### Graph neural networks (GNNs)

Most current GNNs operate according to a message passing paradigm where a representation \( r_v \) is assigned to each node \( v \in \mathcal{V} \) and is updated by \( K \) layers of learnable transformations. These layers first aggregate representations over neighboring nodes \( N(v) \) and then update the current representation via an encoder \( \text{ENC} \). For prevailing GNN models like GCN [18] and GAT [40], each layer can be formalized as \( r_v^{(k)} = \text{ENC}(A^{\text{AGGR}}(r_v^{(k-1)} : u \in N(v) \cup \{v\})) \), where \( A^{\text{AGGR}} \) is the mean function weighted by node degrees (GCN) or an action mechanism (GAT), and \( \text{ENC} \) is a learnable (nonlinear) mapping.

#### Adversarial attacks on graphs

Given a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}) \) and a GNN \( f \) that processes \( \mathcal{G} \), an adversarial attacker tries to create a perturbed graph \( \mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mathcal{X}') \) with a modified edge-set \( \mathcal{E}' \) such that the performance of the GNN \( f \) is maximally degraded. The information available to the attacker can vary under different scenarios [17, 37]. Here, we follow the gray-box formalization by [52], where the attacker knows the training set \( \mathcal{T}_Y \), but not the trained GNN \( f \). The attacker thus considers a surrogate GNN and picks perturbations that maximize an attack loss \( \mathcal{L}_{atk} \) [49, 52], assuming that attacks to the surrogate model are transferable to the attacked GNN. For node classification, the attack loss \( \mathcal{L}_{atk} \) quantifies how the predictions \( z_v \in (0, 1)^{|\mathcal{Y}|} \) made by the GNN \( f \) differ from the true labels. For a targeted attack of node \( v \) with class label \( y_v \in \mathcal{Y} \), we adopt the negative classification margin (CM-type) [42, 52]: \( \mathcal{L}_{atk} = -\Delta_v = -(z_v y_v - \max_{y \neq y_v} z_v y) \). The attacker usually has additional constraints, such as a limit on the size of the perturbations allowed [49, 52].

### Taxonomy of attacks

We follow the taxonomy of attacks introduced in [17, 37]. For node classification, the attacker may aim to change the classification of a specific node \( v \in \mathcal{V} \) (targeted attack), or to decrease the overall classification accuracy (untargeted attack). Attacks can also happen at different stages of the training process: we refer to attacks introduced before training as (pre-training) poison attacks, and attacks introduced after the training process (and before potential retraining on perturbed data) as (post-training) evasion attacks. While our theoretical analysis (§3) mainly considers targeted evasion attacks, we consider other attacks in our empirical evaluation (§5).

### Characterizing homophily and heterophily in graphs

Using class labels, we characterize the types of connections in a graph contributing to its overall level of homophily/heterophily as follows:

- **Definition 1 (Homophilic/Heterophilic Path and Edge).** A \( k \)-hop homophilous path from node \( w \) to \( u \) is a length-\( k \) path between endpoint nodes with the same class label \( y_w = y_u \). Otherwise, the path is called heterophilous. A homophilous or heterophilous edge is a special case with \( k = 1 \).

Following [27, 47], we define the homophily ratio \( h \) as:

- **Definition 2 (Homophily Ratio).** The homophily ratio is the fraction of homophilous edges among all the edges in a graph: \( h = |\{(u, v) \in \mathcal{E} : y_u = y_v\}|/|\mathcal{E}| \).

When the edges in a graph are wired randomly, independent to the node labels, the expectation for \( h \) is \( h_r = 1/|\mathcal{Y}| \) for balanced classes [27]. For simplicity, we informally refer to graphs with homophily ratio \( h \gg 1/|\mathcal{Y}| \) as homophilous graphs (which have been the focus in most prior works), graphs with homophily ratio \( h \ll 1/|\mathcal{Y}| \) as heterophilous graphs, and graphs with homophily ratio \( h \approx 1/|\mathcal{Y}| \) as weakly heterophilous graphs.

## 3 RELATION BETWEEN GRAPH HETEROPHILY & MODEL ROBUSTNESS

In this section, we first show theoretical results on the relation between adversarial structural attacks and the change in the homophily level of the underlying graphs. Though empirical analyses from previous works have suggested this relation on homophilous graphs [17, 41], to our knowledge, we are the first to formalize it with theoretical analysis and address the case of heterophilous graphs. As an implication of the relation, we then discuss how a key design that improves predictive performance of GNNs under heterophily can also help boost their robustness.

### 3.1 How Do Structural Attacks Change Homophily in Graphs?

#### Homophilous graphs: structural attacks are mostly heterophilous attacks

Our first result shows that, for homophilous data, effective structural attacks on GNNs (as measured by loss \( \mathcal{L}_{atk} \)) always result in a reduced level of homophily where either new heterophilous connections are added or existing homophilous connections are removed. It also states that direct perturbations on
1-hop neighbors of the target nodes are more effective than indirect perturbations (influencer attacks [52]) on multi-hop neighbors. For simplicity, akin to previous works [49, 52] we establish our results for targeted evasion (post-training) attacks in a stylized learning setup with a linear GNN. However, our findings generalize to more general setups on real-world datasets as we show in our experiments (§5.1). In the theorems below, we use the notion of gambit node: node \( u \) is called a gambit if a perturbation that targets node \( v \in \mathcal{V} \) adjusts the connectivity of node \( u \in \mathcal{V} \).

**Theorem 1.** Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, X) \) be a self-loop-free graph with adjacency matrix \( A \) and node features \( x_v = p \cdot \text{head}(y_v) + \frac{1-p}{|\mathcal{V}|} \cdot 1 \) for each node \( v \), where 1 is an all-1 vector, and \( p \) is a parameter that regulates the signal to noise ratio. Assume that a fraction \( h \) of each node’s neighbors belong to the same class, while a fraction \( \frac{1-h}{|\mathcal{V}|-1} \) belongs uniformly to any other class. Consider a 2-layer linear GNN \( f^{(k)}(A, X) = \overline{A}^2 X W \) trained on a training set \( \mathcal{D}_v \subseteq \mathcal{D}_V \), with at least one node from each class \( v \in \mathcal{V} \), and degree \( d_v \) for all nodes with a distance less than 2 to any \( v \in \mathcal{D}_V \). For a unit structural perturbation that involves a target node \( v \in \mathcal{D}_v \), and a correctly classified gambit node with degree \( d_a \), the following statements hold if \( h \geq \frac{1}{|\mathcal{V}|}

1. The attack loss \( \mathcal{L}_{atk} (§2) \) of the target \( v \) increases only for actions increasing heterophily, i.e., when removing a homophilous edge or path, or adding a heterophilous edge or path to node \( v \).
2. Direct perturbations on edges (or 1-hop paths) incident to the target node lead to greater increase in \( \mathcal{L}_{atk} \) than indirect perturbations on multi-hop paths to target node \( v \).

We give the proof in App. A. Intuitively, the relative inability of existing GNNs to make full use of heterophilous data [35, 47] can be exploited by inserting heterophilous connections in graphs where homophilous ones are expected. Though the theorem shows that effective attacks on homophilous graphs necessarily reduce the homophily level, the converse is not true: not all perturbations which reduce the homophily level are effective attacks [31].

**Heterophilous graphs: structural attacks can be homophilous or heterophilous, depending on node degrees.** When a graph displays heterophily, our analysis shows a more complicated picture on how the level of homophily in the graph is changed by effective structural attacks: in heterophilic case, the direction of change is dependent on the degrees of both the target node and the gambit node \( u \) of the attack. Specifically, if the degree of either node is low, attacks increasing the heterophily are still effective; however, if the degrees \( d_v \) and \( d_u \) of both nodes are high, attacks decreasing the heterophily will be effective. Similar to the homophilous case, we formalize our results below for targeted evasion attacks in a stylized learning setup.

**Theorem 2.** Under the setup of Thm. 1, for a unit perturbation that involves a target node with degree \( d_v \), and a correctly classified gambit node with degree \( d_u \), the following statements hold:

1. **(Low-degree target node)** if \( 0 < d_v < |\mathcal{V}| - 2 \), for any \( d_u \geq 0 \) and \( h 
\)

2. **(High-degree target node)** if \( d_v > |\mathcal{V}| - 2 \), conditioning on the degree \( d_u \) of the gambit node:

   a. **(Low-degree gambit node)** if \( d_u < \frac{(d_v+2)(|\mathcal{V}|-1)}{d_v-1} \), for any \( h \in [0, 1] \), the attack loss \( \mathcal{L}_{atk} (§2) \) of \( v \) increases only under actions increasing heterophily in the graph.

   b. **(High-degree gambit node)** if \( d_u \geq \frac{(d_v+2)(|\mathcal{V}|-1)}{d_v-1} \), for \( 0 \leq h < \frac{d_v(d_v-2)(|\mathcal{V}|+2)-d(d_v+2)(|\mathcal{V}|-1)}{d_v(d_v+1)|\mathcal{V}|^2} < \frac{1}{|\mathcal{V}|} \), \( \mathcal{L}_{atk} (§2) \) of \( v \) increases only under actions reducing heterophily.

In the statements above, the actions increasing heterophily include removing a homophilous edge or adding a heterophilous edge to node \( v \), and the actions reducing heterophily include adding a homophilous edge or removing a heterophilous edge to node \( v \).

The above theorems cover the situation when the gambit nodes are initially classified correctly (where attacks introducing heterophily can be unambiguously defined using the ground-truth class labels of the nodes involved). However, in §5.1, we show on real-world datasets that a relaxed interpretation of the theorems, where heterophily is instead defined by the predicted class labels of GNNs, can explain the behavior of the attacks regardless of the initial correctness of the gambits.

**3.2 Boosting Robustness with A Simple Heterophilous Design**

A natural follow-up question is whether GNNs with better performance under heterophily are also more robust against structural attacks. We deduce that a key design for improving GNN performance for heterophilous data—separate aggregators for ego- and neighbor-embeddings—can also boost the robustness of GNNs by enabling them to better cope with adversarial changes in heterophily.

**Separate aggregators for ego- and neighbor-embeddings.** This design uses separate GNN aggregators for ego-embedding \( \tau_u \) and neighbor-embeddings \( \{\tau_u : u \in N(v)\} \). Formally, the representation learned for node \( v \) in the \( k \)-th layer is:

\[
\begin{align*}
\nu^{(k)}_v &= \text{ENC} \left( \text{AGGR1}(\nu^{(k-1)}_u : u \in N(v)) \right), \\
\nu^{(0)}_v &= \text{AGGR2}(\nu^{(k-1)}_u : u \in N(v))
\end{align*}
\]

where AGGR1 and AGGR2 are separate aggregators, such as averaging functions (GCN), attention mechanisms (GAT), or other pooling mechanisms [15]. This design has been utilized in existing GNN models (we show examples later in this section), and has been shown to significantly boost the representation power of GNNs under natural heterophily [47]. The ego-aggregator AGGR1 may also introduce skip connections [43] to the ego-embeddings aggregated in previous layers as shown in Eq. (1), which is another design that further improves the representation power under heterophily [47].

**Intuition.** The key design changes, as compared to the GCN formulation in §2, allow for the ego-embedding \( \tau_u \) to be aggregated and weighted separately from the neighbor-embeddings \( \{\tau_u : u \in N(v)\} \), as well as for the use of skip connections to ego-embeddings of previous layers. Intuitively, ego-embeddings of feature vectors at the first layer are independent of the graph structure and thus unaffected by adversarial structural perturbations. Hence, a separate aggregator and skip connections can provide better access to unperturbed information and mitigate the effects of the attacks.

**Theoretical analysis.** We formalize the above intuition that shows how separate aggregators for ego- and neighbor-embeddings enable GNN layers to reduce the attack loss.

**Theorem 3.** Under the setup of Thm. 1, consider two alternative layers from which a two-layer linear GNN is built: (1) a layer designed as \( f(A, X) = \overline{A}^2 X W \); and (2) a layer formulated as \( f(A, X; \alpha) = ((1-\alpha)A + \alpha I) X W \), which mixes the ego- and neighbor-embedding linearly under a predefined weight \( \alpha \in [0, 1] \). Then, for \( h > 1/|\mathcal{V}| \),
\( \alpha > 1/(1 + d_a) \), and a unit perturbation increasing \( L_{atk} \) as in Thm. 1, outputs of layer \( f \) lead to a strictly smaller increase in \( L_{atk} \) than \( f_e \).

We provide the proof in App. A; note that for \( \alpha = 1/(1 + d_a) \), the two layers are the same: \( f(A,X;\alpha) = f_e(A,X) \). Theorem 3 shows that an increase to the weights of ego-embedding (manually or through training) improves the robustness of the GNN \( f \) for a homophily ratio \( h > 1/|Y| \). Though aggregators and encoders are stylized in this simple instantiation of the design in the theorem, the empirical analysis in §5.2 confirms that GNNs with more advanced aggregators and encoders, which we will discuss next, also benefit from separate aggregators. Specifically, we find that such GNNs outperform methods without this design by up to 40.00% and 48.88% on homophilous and heterophilous graphs, respectively, while performing comparably on clean datasets.

**Instantiations of the design on GNNs.** We demonstrate how the heterophilous design outlined in Eq. (1) is instantiated in various GNN models, which are used in our empirical evaluation in §5. In particular, we highlight how these GNN architectures allow separate aggregations of the ego- and neighbor-embeddings.

- **In \( H_2 \)GCN [47],** a final representation is computed for each node \( v \in V \) through \( r_v^{(\text{final})} = \text{CONCAT}(r_v^{(0)}, r_v^{(1)}, \ldots, r_v^{(K)}) \), where \( r_v^{(k)} \) is the non-linear ego-embedding of node features and \( r_v^{(k)} \) are the intermediate representations aggregated in the \( k \)-th layer, where \( k \in (1, ..., K) \). By interpreting the update rule’s CONCAT as the ENC operation, \( AGGR_1 \) as the skip connection to the ego-embedding of node features, and the concatenation of the intermediate representations as \( AGGR_2 \), the ego- and neighbor-embeddings are separately aggregated as stated in the design.

- **GraphSAGE** (with mean aggregator) [15] utilizes a concatenation-based encoding scheme through their update of \( r_u^{(k)} = \sigma \left( \text{CONCAT} \left( \{ r_v^{(k)} \}_{v \in N(i)} \right) \cdot W \right) \), where ENC \( \{ x_1, x_2 \} = \sigma(\text{CONCAT}(x_1, x_2) \cdot W) \), \( AGGR_1 \) as \( r_u^{(k-1)} \), and \( AGGR_2 \) as the mean function.

- **GPR-GNN** [8] embeds each node feature vector separately with a fully connected layer to compute \( R_v^{(0)} \) (or \( H_v^{(0)} \) as in the original paper), similar to \( H_2 \)GCN, and then updates each node’s hidden representations through a weighted sum of all \( k \)-th hop layers around the ego-node, where \( k \in (0, 1, ..., K) \). By interpreting the summation as the ENC operation, \( AGGR_1 \) as \( \gamma R_v^{(0)} \), and \( AGGR_2 \) as \( \sum_{k=1}^{K} \gamma_k \hat{A}^{k} R^{(k-1)} \), where \( \gamma \) denotes the weights associated with each \( k \)-hop ego network, the aggregation of the ego- and neighbor-embeddings is decoupled.

- **FAGCN** [2] follows a similar update function to GPR-GNN with \( r_i^{(k)} = r_i^{(0)} + \sum_{j \in N(i)} \frac{a_{ij}^{(l-1)}}{\sqrt{d_id_j}} r_j^{(l-1)} \), where \( r_i^{(0)} \) (or \( h_i^{(0)} \) in the original paper) represents the non-linear ego-embedding and \( a_{ij}^{(l-1)} \) is a constant measuring the ratio of low and high frequency components. The heterophilous design can similarly be recovered by interpreting the sum as the ENC operation, \( AGGR_1 \) as \( r_i^{(0)} \) as a weighted skip connection to the ego-embedding of features, and the weighted sum of embeddings within the neighborhood \( N(i) \) of node \( i \in V \) as \( AGGR_2 \).

- **CPGNN** [48] formulates the update function of belief vectors \( R^{(k)} \) after the \( k \)-th propagation layer as \( R^{(k)} = R^{(0)} + AR^{(k-1)}H \), where \( R^{(0)} \) (in the original paper) consists of prior belief vectors for each node (as the ego-embeddings \( r_i^{(0)} \) in Eq. (1)), and \( H \) is the learnable compatibility matrix. The heterophilous design is recovered by letting \( AGGR_1(\cdot) = R^{(0)} \) as a skip connection, \( AGGR_2(\cdot) = AR^{(k-1)}H \), and the ENC operation as the summation.

### 4 RELATED WORK

**Adversarial attacks and defense strategies for graphs.** Since **Nettack** [52] and **RL-S2V** [10] first demonstrated the vulnerabilities of GNNs against adversarial perturbations, a variety of attack strategies under different scenarios have been proposed, including adversarial attacks on the graph structure [3, 7, 10, 24, 42], node features [30, 38], or combinations of both [41, 49, 52]. On the defense side, various techniques for improving the GNN robustness against adversarial attacks have been proposed, including: adversarial training [4, 42, 49]; **RGCN** [46], which adopts Gaussian-based embeddings and a variance-based attention mechanism; low-rank approximation of graph adjacency [12] against Nettack [52]; **ProGNN** [16], which estimates the unperturbed graph structure in training with the assumptions of low-rank, sparsity, and homophily of node features; **GCN-Jaccard** [41] and **GNNGuard** [45], which assume homophily of features (or structural embeddings) and train GNN models on a pruned graph with only strong homophilous links; and **Soft Medoid** [13], an aggregation function with improved robustness. Other recent works have looked into the certification of nodes that are guaranteed to be robust against certain structural and feature perturbations [4, 50, 51], including approaches based on model-agnostic randomized smoothing [5, 9, 21]. Interested readers can refer to the recent surveys [17, 37] for a comprehensive review.

**GNNs & Heterophily.** Recent works [28, 31, 35, 47] have shown that heterophilous datasets can lead to significant performance loss for popular GNN architectures (e.g., **GCN** [18], **GAT** [40]). This issue is also known in classical semi-supervised learning [34]. To address this issue, several GNN designs for handling heterophilous connections have been proposed [1, 2, 11, 25, 35, 47, 48]. Yan et al. [44] recently discussed the connection between heterophily and oversmoothing for GNNs, and designs to address both issues; [29] studied how locally-occurring heterophily affects fairness of GNNs. However, the formal connection between heterophily and robustness of GNNs has received little attention. Here we focus on a simple yet powerful design that significantly improves performance under heterophily [47], and can be readily incorporated into GNNs.

### 5 EMPIRICAL EVALUATION

Our analysis seeks to answer the following questions: (Q1) Does our theoretical analysis on the relations between adversarial attacks and changes in heterophily level generalize to real-world
datasets? (Q2) Do heterophily-adjusted GNNs, i.e., models with separate aggregators for ego- and neighbor-embeddings, show improved robustness against state-of-the-art attacks? (Q3) Does the identified design improve the certifiable robustness of GNNs?

First, we describe the experimental setup and datasets that we use to answer the above questions.

**Attack Setup.** We consider both targeted and untargeted attacks (§2), generated by Nettack [52] and Metattack [49], respectively. For each attack method, we consider poison (pre-training) and evasion (post-training) attacks, yielding 4 attack scenarios in total. We focus on robustness against structural perturbations and keep the node features unchanged. We randomly generate 3 sets of perturbations per attack method and dataset, and consistently evaluate each GNN model on them. For Nettack, we randomly select 60 nodes from the graph as the target nodes for each set of perturbations, instead of the GCN-based target selection approach as in [52]: the approach in [52] only selects nodes that are correctly classified by GCN [18] on clean data; this introduces unfair advantage towards GCN, especially on heterophilous datasets where GCN can exhibit significantly inferior accuracy to models like GraphSAGE [47]. For the experiments in §5.1, we use a budget of 1 perturbation per target node to match the setup of our theorems; for the benchmark study in §5.2, we use an attack budget equal to a node’s degree and allow direct attacks on target nodes. For Metattack, we budget the attack as 20% of the number of edges in each dataset, and we use the Meta-Self variant as it shows the most destructiveness [49].

**GNN Models.** To show the effectiveness of our identified design, we evaluate four groups of models against adversarial attacks: (1) Baseline models without any vaccination, including some of the most popular methods: GCN [18], GAT [40], and the graph-agnostic multilayer perceptron (MLP) which relies only on node features; (2) State-of-the-art “vaccinated” baselines designed with robustness in mind: ProGNN [16], GNNGuard [45], GCN-SVD [12] and GCN-SMGDC, which adopts the Soft Medoid aggregator [13] and GDC [20] on GCN [18] architecture; (3) Models with the heterophilic design only: GraphSAGE [15], H$_2$GCN [47], CPGN [48], GPR-GNN [8] FAGCN [2] and APPNP [19]: we discussed how these models instantiate this design in §3.2; (4) Models with both the heterophilic design and explicit robustness-enhancing mechanisms, where we adopt two existing mechanisms: (i) SVD-based low-rank approximation [12] (H$_2$GCN-SVD and GraphSAGE-SVD), and (ii) Soft Medoid aggregator [13] with GDC [20] (H$_2$GCN-SMGDC and GraphSAGE-SMGDC). We combine both these mechanisms with heterophily-adjusted GNNs instead of non-adjusted models (e.g., GCN)—detailed formulations are given on our repository. We set the number of layers as 2 and the size of hidden units per layer as 64 for all models to ensure a fair comparison between different architectures and designs. We provide more implementation details and hyperparameter settings on our repository (App. §B).

**Datasets & Evaluation Setup.** We consider three widely-used citation networks [33, 36] with strong homophily—Cora [32], Pubmed, and Citeseer—along with one weakly and one strongly heterophilous graph, introduced by Lim et al. [27]: FB100 [39] and Snap Patents [22, 25]. We report summary statistics in Table 1, and provide more details on our repository. For computational tractability, we sample the Snap Patents data via snowball sampling [14], where we keep 20% of the neighbors for each traversed node; we give detailed algorithm on our repository. The sizes of the datasets that we used in our experiments are similar to those in previous works on GNN robustness [13, 16]. We follow the evaluation procedure of [16, 52] to split the nodes of each dataset into training (10%), validation (10%) and test (80%) data, and determine the model parameters on training and validation splits. We report the average performance and standard deviation on the 3 sets of generated perturbations. For targeted attacks with Nettack, we report the classification accuracy on the target nodes; for untargeted attacks with Metattack, we report it over the whole test dataset.

**Robustness Certificates.** We adopt randomized smoothing for GNNs [5] to evaluate the certifiable robustness, with parameter choices detailed in our GitHub repository. We only consider structural perturbations in the randomization scheme. Following Geisler et al. [13], we measure the certifiable robustness of GNN models with the accumulated certifications (AC) and the average maximum certifiable radii for edge additions ($r_a$) and deletions ($r_d$) over all correctly predicted nodes. More specifically, AC is defined as $-R(0,0)+\sum_{r_a\geq 0, r_d\geq 0} R(r_a, r_d)$, where $R(r_a, r_d)$ is the certifiably correct ratio, i.e., the ratio of the nodes in the test splits that are both predicted correctly by the smoothed classifier and certifiably robust at radius $(r_a, r_d)$. In addition, we report the accuracy of each model with randomized smoothing enabled on the test splits of the clean datasets, which is equal to $R(0,0)$. We report the average and standard deviation of each statistic over the 3 different training, validation and test splits.

**Hardware Specifications.** We use a workstation with a 12-core AMD Ryzen 9 3900X CPU, 64GB RAM, and a Quadro P6000 GPU with 24 GB GPU Memory.

**Code and Additional Details.** Code and additional details on the setups and results are available on GitHub repository: https://github.com/GemsLab/HeteRobust.

### 5.1 (Q1) Structural Attacks are Mostly Heterophilous: Empirical Validation

To show that our theoretical analysis in §3.1 generalizes to more complex settings beyond the assumptions we made in the theorems, we look into effective targeted attacks made by Nettack on real-world homophilous and heterophilous datasets, and present statistics of the attacks in Table 2, with a focus on the ratios of heterophilic attacks. We use a budget of 1 perturbation per target node in this experiment, and the statistics are reported among all effective perturbations targeting nodes that are correctly classified on clean datasets by the surrogate GNN of Nettack (i.e., GCN) as described in §5. To validate the dependency between the degrees of the target/gambit nodes and the changes of heterophily predicted by Thm. 2, we also show the scatter plots of node degrees in Fig. 1.
Table 2: Effective targeted attacks by NetTack ($\S$5.1): ratios of edge additions, deletions and heterophilous attacks (i.e., attacks increasing heterophily). We consider two heterophily definitions, one based on ground-truth class labels (Label), and the other on predicted class labels by GCN on clean datasets (Pred.). All attacks are direct perturbations on edges incident to the targets. Degrees of target and gambit nodes in the attacks are shown in Fig. 1. All attacks introduce heterophilous edges that connect nodes with different predicted labels, following the takeaways of Thm. 1 and 2.

| Dataset | Sample Sizes | Attack Type | Hete. Attacks |
|---------|--------------|-------------|---------------|
|         |              | Add. | Del. | Label | Pred. |
| Cora    | 150          | 99.33%| 0.67%| 100.00%| 100.00%|
| Pubmed  | 153          | 100.00%| 0.00%| 100.00%| 100.00%|
| Citeseer| 121          | 100.00%| 0.00%| 100.00%| 100.00%|
| FB100   | 112          | 100.00%| 0.00%| 50.00% | 100.00%|
| Snap    | 51           | 100.00%| 0.00%| 64.71% | 100.00%|

Figure 1: Scatter plots of the degrees of the target nodes (x-axis) and gambit nodes (y-axis) involved in the targeted attacks ($\S$5.1). Attacks tend to leverage gambit nodes with low degrees, which makes attacks increasing heterophily effective for heterophilous graphs following Thm. 2.

Homophilous Networks. For the strongly homophilous Cora, Pubmed and Citeseer graphs, all changes introduced by effective attacks in the graph structure follow the conclusion of Thm. 1: they reduce homophily (increase heterophily) by adding heterophilous edges or removing homophilous edges. These results show that despite the simplified analysis, the takeaway of Thm. 1 can be generalized to real-world datasets. In addition, the attacks mostly introduce, rather than prune, edges, suggesting that attacks adding outlier edges to the graph are more powerful than attacks removing informative existing edges. These observations in our experiments are consistent with the observations from previous works [13, 17].

Heterophilous Networks. For heterophilous graphs FB100 ($h \approx 1/|V|$) and Snap ($h < 1/|V|$), Fig. 1 shows that almost all attacks leverage gambit nodes with low degrees (1 or 2); no node with degree higher than 5 is leveraged. All attacks leveraging correctly classified gambit nodes are connecting node $u \in V$ with a different ground-truth class label $y_u \neq y_v$ to the target nodes $v \in V$; attacks leveraging incorrectly classified gambit nodes are always connecting node $u$ with a different predicted class label $\hat{y}_u \neq \hat{y}_v = y_v$ to the target node $v$, even though some gambit nodes have the same ground-truth class label $y_u = y_v \neq \hat{y}_u$ as the target nodes. These results validate the conclusion of Thm. 2 on correctly classified gambit nodes, and demonstrate its generalizability under the heterophily definition based on predicted class labels. Note that the predicted class labels $\hat{y}_u$ for each node $u \in V$ are based on GCN, which is the surrogate GNN used by NetTack.

5.2 (Q2) Benchmark Study of GNN Models: Heterophilous Design Leads to Improved Empirical Robustness

To answer (Q2) on whether heterophily-adjusted GNN models show improved performance against state-of-the-art attacks, we conduct a comprehensive benchmark study. We consider all four categories of GNN models mentioned in $\S$, and evaluate their robustness against both targeted and untargeted attacks. We report the hyperparameters for each method on our repository ($\S$B.3). Table 3 shows the performance of each method under poison (pre-training) attacks and on clean (uncerturbed) data, and Fig. 2 visualizes the corresponding performance changes relative to the clean datasets. For conciseness, we report additional results on Pubmed in Table 5, and under evasion (post-training) attacks on our GitHub repository (Table 7 and 8), where we also discuss how our simple heterophilous design leads to only minor computational overhead compared to existing vaccination mechanisms ($\S$C.4).

Targeted attacks by NetTack. (1) Poison attacks. Under targeted poison attacks, Table 3 (left) shows that GraphSAGE-SVD and H2GCN-SVD, which combine our identified design with a low-rank vaccination approach adopted in GCN-SVD [12], outperform state-of-the-art vaccinated methods across all datasets by up to 13.34% in homophilous settings and 18.33% in heterophilous settings. Furthermore, GraphSAGE-SMGC and H2GCN-SMGC, which combine our design with existing vaccinations based on Soft Medoid [13] and GDC [20], show better performance against attacks in all datasets compared to GCN-SMGC, the corresponding baseline without our design, with up to 19.44% improvement on homophilous settings and 30.55% improvement on heterophilous settings. In summary, these observations show that the heterophilous design improves the robustness of GNNs alongside existing vaccination mechanisms.

Methods merely employing the identified design also show significantly improved robustness, though there are differences in the amount of robustness improvement due to architectural differences. Specifically, these methods outperform the best unvaccinated method (GAT) on all datasets by up to 33.75% in average, despite having mostly comparable performance on clean datasets; methods like APPNP and CPGN also show comparable or even better robustness than state-of-the-art vaccinated GNNs. These observations also apply to the larger Pubmed dataset in Table 5. We also note that the graph-agnostic MLP, which is immune to structural attacks, outperforms all GNNs against attacks on Citeseer and Snap; this shows the challenges in defending against targeted attacks and calls for more effective defense strategies upon our discoveries.

(2) Evasion attacks. Under evasion attacks (detailed results are reported in App. Table 7 on our repository), we observe similar trends.
How does Heterophily Impact the Robustness of Graph Neural Networks? Theoretical Connections and Practical Implications

Table 3: Benchmark study: mean accuracy ± stdev against poison attacks, with accuracy on clean datasets in gray for reference. Accuracy is reported on target nodes for NETTACK, and on full test splits for Metattack. Best GNN performance against attacks is highlighted in blue per dataset, and in gray per model group. MLP is immune to structural attacks and not considered as a GNN model. Accuracy against evasion attacks are listed on our GitHub repository (App. §C.1), and the setups in §5. Additional results on Pubmed are listed in App. Table 5. GNNs merely adopting this design achieve up to 40.00% improvement in accuracy against NETTACK compared to the best-performing unvaccinated model (GCN). Additionally, methods combining this design alongside explicit defense mechanisms (e.g., GraphSAGE-SVD) achieve further robustness improvement to the corresponding base mechanism without the design (e.g., GCN-SVD), and outperform the best vaccinated baseline by up to 18.33%.

![Table 3](image)

Figure 2: (Best viewed in color.) Classification accuracy on clean data and against poison attacks for target nodes attacked by NETTACK. Error bars show standard deviation across different sets of experiments. Detailed results are listed in Table 3. As expected, MLP is not influenced by the adversarial structural attacks.

as in poison attacks: GraphSAGE-SVD and H2GCN-SVD are up to 20.55% more accurate than the GCN-SVD, the corresponding baseline without the heterophilous design, and GraphSAGE-SMGDC and H2GCN-SMGDC outperform GCN-SMGDC by up to 19.44%. Methods featuring the identified design alone achieve up to 38.89% gain in average performance against the best unvaccinated baseline, which we also observe on Pubmed. We note that two baselines, GNNGuard and ProGNN, are designed specifically to defend against poison attacks, and are not capable of addressing evasion attacks.

Un targeted attacks by Metattack. 1 Poison attacks. We also test the robustness of each method against untargeted attacks. Table 3 (right) shows the performance under poison attacks. Though our theoretical analysis in §3 focuses on the effect of the heterophilous design under targeted attacks, we observe similar improvements in robustness against untargeted attacks in the poison setup. GNNs with the identified design show mostly improved robustness compared to unvaccinated models, while having similar performance on the clean datasets. Specifically, CPGNN shows exceptional robustness, outperforming the best unvaccinated model by up to 32.85%. Moreover, models combining the identified design with low-rank approximation show up to 21.04% improvement in accuracy compared to GCN-SVD, which uses only low-rank approximation. Models combining the design with Soft Medoid and GDC show up to 37.29% improvement in accuracy compared to GCN-SMGDC. We also note that the most robust method for each dataset is among the ones with the identified design. These results again support the effectiveness of the heterophilous design in boosting the robustness of GNNs in addition to existing vaccination mechanisms.

2 Evasion attacks. We present the performance under evasion attacks on our GitHub repository. Unlike the poison attacks, the evasion setup only leads to a slight decrease in average accuracy of less than 2% for most models. Moreover, there appears to be no clearly increased robustness for vaccinated models (with the identified design or other vaccination mechanisms) compared to unvaccinated models. This can be attributed to the reduced effectiveness of evasion vs. poison attacks (as in NETTACK), and the increased challenges of untargeted attacks.

2643
Table 4: Accumulated certifications (AC), average certifiable radii \((r_a\) and \(r_d\)) and accuracy of GNNs with randomized smoothing enabled (i.e., \(f(\phi(s))\)) on the test splits of the clean datasets, with randomization schemes \(\phi\) allowing both addition and deletion (i.e., \(p_a = 0.001, p_d = 0.4\)), and additional only (i.e., \(p_a = 0.001, p_d = 0\)). For each statistic, we report the mean and stdv across 3 runs. Best results highlighted in blue per dataset, and in gray per model group. We provide results with the deletion only scheme on our repository (App. §C.2). APPNP, with the identified design, improves the accumulated certification (AC) by up to 5.3x on homophilous datasets and 10.1x on heterophilous ones compared to the best performing baseline without the design.

| Hete. GNN | Add. & Del. | AC | \(r_a\) | \(r_d\) | Acc. % | AC | \(r_a\) | \(r_d\) | Acc. % |
|-----------|-------------|----|---------|---------|--------|----|---------|---------|--------|
| FAGCN     | ✓           | 3.90 ± 0.48 | 3.40 ± 0.56 | 79.34 ± 0.77 | 0.42 ± 0.02 | 0.53 ± 0.03 | 80.97 ± 0.76 | 2.92 ± 0.03 | 3.27 ± 0.04 | 71.44 ± 0.69 | 0.29 ± 0.01 | 0.40 ± 0.02 | 72.99 ± 0.92 |
| GraphSAGE | ✓           | 2.10 ± 0.13 | 2.43 ± 0.57 | 79.61 ± 0.28 | 0.28 ± 0.03 | 0.34 ± 0.03 | 81.07 ± 0.67 | 2.21 ± 0.19 | 2.56 ± 0.02 | 73.84 ± 0.48 | 0.33 ± 0.01 | 0.44 ± 0.01 | 74.70 ± 0.76 |
| CPGNN     | ✓           | 1.87 ± 0.14 | 2.24 ± 0.57 | 75.37 ± 0.37 | 0.17 ± 0.00 | 0.21 ± 0.08 | 78.34 ± 0.76 | 2.03 ± 0.11 | 2.52 ± 0.29 | 73.48 ± 0.37 | 0.15 ± 0.02 | 0.20 ± 0.04 | 74.62 ± 0.30 |
| GPR-GNN   | ✓           | 4.42 ± 0.63 | 4.35 ± 0.67 | 74.90 ± 0.49 | 0.43 ± 0.00 | 0.55 ± 0.00 | 76.96 ± 0.69 | 4.63 ± 0.81 | 4.92 ± 0.00 | 66.33 ± 0.23 | 0.40 ± 0.00 | 0.59 ± 0.00 | 67.52 ± 0.40 |
| APPNP     | ✓           | 4.30 ± 0.57 | 4.25 ± 0.49 | 76.49 ± 0.78 | 0.43 ± 0.00 | 0.54 ± 0.00 | 79.04 ± 0.79 | 4.07 ± 0.58 | 4.23 ± 0.00 | 71.82 ± 0.75 | 0.38 ± 0.00 | 0.53 ± 0.00 | 72.41 ± 0.76 |
| FAGCN     | ✓           | 10.11 ± 1.86 | 8.52 ± 0.71 | 71.97 ± 0.71 | 0.48 ± 0.00 | 0.95 ± 0.00 | 72.72 ± 1.00 | 9.87 ± 1.00 | 1.38 ± 1.34 | 69.39 ± 0.79 | 0.66 ± 0.95 | 0.97 ± 1.00 | 69.41 ± 0.76 |

Table 5: Additional results on Pubmed (details in App. §C.1).

| Hete. GNN | Variants | Poison | Evasion | Clean |
|-----------|----------|--------|---------|-------|
| H2GCN     | SVD      | 44.44 ± 3.75 | 46.67 ± 3.76 | 78.78 ± 5.77 |
| GraphSAGE | SVD      | 33.33 ± 3.75 | 34.44 ± 3.75 | 84.44 ± 5.77 |
| CPGNN     |          | 60.00 ± 3.75 | 60.00 ± 3.75 | 82.78 ± 5.77 |
| GPR-GNN   |          | 13.89 ± 3.75 | 15.56 ± 3.75 | 85.56 ± 5.77 |
| FAGCN     |          | 27.78 ± 3.75 | 31.67 ± 3.75 | 86.67 ± 5.77 |
| APPNP     |          | 79.44 ± 3.75 | 81.67 ± 3.75 | 86.67 ± 3.75 |
| GNNGuard  |          | 73.89 ± 3.75 | 69.17 ± 3.75 | 82.78 ± 5.77 |
| GAT       |          | 5.56 ± 3.75 | 5.56 ± 3.75 | 85.00 ± 3.75 |
| MLP       |          | 86.11 ± 3.75 | 86.11 ± 3.75 | 86.11 ± 3.75 |

5.3 (Q3) Heterophily-adjusted GNNs are Certifiably More Robust

It is worth noting that robustness against specific attacks such as NETTACK and Metattack does not guarantee robustness towards other possible attacks. To overcome this limitation, robustness certificates provide guarantees (in some cases probabilistically) that attacks within a certain radius cannot change a model’s predictions. Complementary to our evaluation on empirical robustness, we further demonstrate that heterophily-adjusted GNNs featuring our identified design are certifiably more robust than methods without it, thus answering (Q3). For GNN models and datasets, we exclude the larger Pubmed dataset and models that learn to rewrite the graph structure through the training process, or require recalculation of the low-rank approximation or inverse of matrices (as used by GDC [20]) for every randomized perturbation, as we find that sampling on these setups is computationally challenging. We use the same hyperparameters as the benchmark study in §5.2.

Table 4 shows multiple metrics of certifiable robustness of each GNN model under edge randomization schemes allowing both addition and deletion, and allowing addition only; we additionally report results under a scheme allowing only deletion on our GitHub repository. For the scheme allowing both addition and deletion, we observe that all heterophily-adjusted methods have better certifiable robustness compared to methods without the design. Specifically, on homophilous datasets (Cora and Citeseer), methods with the identified design achieve an up to 5.3 times relative improvement in accumulated certification. On heterophilous datasets (FB100 and Snap), they outperform the baselines by a factor of 11.1. In the more challenging case with the addition only scheme, methods with the design also show up to 2.9 times relative increase in AC on the homophilous datasets and 11.0 times relative increase in AC on the heterophilous datasets compared to the baselines. For the deletion only scheme, we find that unvaccinated models like GCN already have decent certifiable robustness in this scenario, commensurating with our discussions in §5.1 that deletions create less severe perturbations. Overall, our results show that models featuring our identified design achieve significantly improved certifiable robustness compared to models lacking this design. However, like in our empirical robustness evaluation, architectural differences lead to some variability of robustness; the results also show tradeoffs between accuracy and robustness. We also observe that the rankings under certifiable and empirical robustness are different, as in the previous results from [13], we discuss more in our repository.

6 CONCLUSION

We formalized the relation between heterophily and adversarial structural attacks, and showed theoretically and empirically that effective attacks gravitate towards increasing heterophily in both homophilous and heterophilous graphs by leveraging low-degree (gambit) nodes. Using these insights, we showed that a key design addressing heterophily, namely separate aggregators for ego- and neighbor-embeddings, can lead to competitive improvement on empirical and certifiable robustness, with only small influence on clean performance. Finally, we compared the design with state-of-the-art vaccination mechanisms under different attack scenarios for
various datasets, and illustrated that they are complementary and that their combination can lead to more robust GNN models. We note that while we focus on the structural attacks, GNNs are also vulnerable to other types of attacks such as feature perturbations. We hope our analysis can inspire more effective defense strategies against adversarial attacks, especially designs that improve robustness by better addressing heterophily, such as heterophily in node features, or locally-occurring heterophily in homophilous graphs.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under CAREER Grant No. IIS 1845491 and Medium grant, Army Young Investigator Award No. W911NF1810397, an Adobe Digital Experience research faculty award, an Amazon faculty award, a Google faculty award, and AWS Cloud Credits for Research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research. MTS received funding from the Ministry of Culture and Science (MKW) of the German State of North Rhine-Westphalia (“NRW Rückkehrprogramm”) and the Excellence Strategy of the Federal Government and the Länder. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] Sami Abu-El-Hajja, Bryan Perozzi, Amol Kapoor, Hravy Harutyunyan, Nazanin Alipourfarid, Kristina Lerman, Greg Ver Steeg, and Aram Galstyan. MixHop: Higher-order graph convolution architectures via sparsified neighborhood mixing. In ICML, 2019.
[2] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In AAAI, 2021.
[3] Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In ICLR, 2019.
[4] Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. In NeurIPS, 2019.
[5] Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. Efficient robustness certificates for discrete data. Sparsity-aware randomized smoothing for graphs, images and more. In ICLR, 2020.
[6] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine, 34(4):18–42, 2017.
[7] Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Wenzhu Zhou, and Junzhou Huang. A restricted bilevel adversarial framework towards attacking graph embedding models. In AAAI, 2020.
[8] Eli Chien, Jianhao Peng, Pan Li, and Olga Milenkovic. Adaptive universal generalized pagerank graph neural network. In ICRL, 2021.
[9] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In ICML, pp. 1310–1320. PMLR, 2019.
[10] Hanjun Dai, Hui Li, Han Xu, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Graph neural networks with adaptive frequency response filter. In ICLR, 2021.
[11] Tsubasa Takahashi. Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. In Big Data, IEEE, 2019.
[12] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. Physica A: Statistical Mechanics and its Applications, 391(16): 4165–4180, 2012. ISSN 0378-4371. doi: 10.1016/j.physa.2011.12.021.
[13] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. arXiv preprint arXiv:1710.10903, 2017.
[14] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Antti Rasch=scia. Graph convolutional networks on hypergraphs. In KDD, 2020.
[15] Meng Liu, Zhengyang Wang, and Shuiwang Ji. Non-local graph neural networks. In NeurIPS, 2018.
[16] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. Towards more practical adversarial attacks on graph neural networks. In NeurIPS, 2020.
[17] Jiong Zhu, Ryan A Rossi, Amin Afshar, Zico Kolter, and Danai Koutra. Graph neural networks against adversarial attacks. In NeurIPS, 2018.
SUPPLEMENTARY MATERIAL ON REPRODUCIBILITY

For reproducibility, we describe main experimental setups at the beginning of §5, including the attack configurations, the depth and hidden unit size for GNNs, ratios for training, validation and test splits, and hardware specifications. We provide code and datasets on GitHub repository: https://github.com/GemsLab/HeteRobust, where we also include additional details on the setups and results, including the implementations and detailed hyperparameters for GNNs and randomized smoothing (§B), and additional results for evasion attacks (§C.1) and certifiable robustness (§C.2). In addition, we report results for GNN models against NETTACK on the larger Pubmed dataset in Table 5, with more details in §C.1: GCN-SVD and SMGDC-based vaccinations run out of memory on Pubmed.

A PROOFS AND DISCUSSIONS OF THEOREMS

For simplicity of mathematical expressions, we use $C_{i,j}(s,t)$ to refer a matrix where all elements in the $i$-th column are $s$, with all remaining elements (not in the $i$-th column) as $t$; we use $e_i(s,t)$ when the matrix is a row vector. We further denote $C^*(s,t)$ as a circulant matrix with all diagonal elements as $s$ and all off-diagonal elements as $t$, and $B(s,t)$ as block matrix of the following form:

$$B(s,t) = \begin{bmatrix}
C_1^*(s,t) & C_2^*(s,t) & \cdots & C_{|Y|}^*(s,t)
\end{bmatrix}^T$$

Proof for Thm. 1. We give the proof in three parts: first, we analyze the training process of the GNN $f^{(2)}_s(A,X) = \tilde{A}_s^2XW$ on clean data and analytically derive the optimal weight matrix $W_*$ in a stylized learning setup; then, we construct a targeted evasion attack and calculate the attack loss for a unit structural attack; last, we summarize and validate the statements in the theorem.

Stylized learning on clean data. Given the 2-layer linearized GNN $f^{(2)}_s(A,X) = \tilde{A}_s^2XW$ and the training set $\mathcal{T}_V \subseteq \mathcal{D}_V$, the goal of the training process is to optimize the weight matrix $W$ to minimize the cross-entropy loss function $L([z]_{\mathcal{F}_V}, [Y]_{\mathcal{F}_V})$, where predictions $[z]_{\mathcal{F}_V} = [\tilde{A}_s^2X]_{\mathcal{F}_V}W$ correspond to the predicted class label distributions for each node $v$ in the training set $\mathcal{F}_V$, and $[Y]_{\mathcal{F}_V}$ is the one-hot encoding of class labels provided in $\mathcal{T}_V$.

Without loss of generality, we reorder $\mathcal{T}_V$ accordingly such that the one-hot encoding of labels for nodes in the training set $[Y]_{\mathcal{F}_V} = B(1,0)$ is in increasing order of the class label $y$. Now we look at the term $[\tilde{A}_s^2X]_{\mathcal{F}_V} = [\tilde{A}_s^2X]_{\mathcal{F}_V}W$, which are the feature vectors aggregated by the two GNN layers for nodes $v$ in the training set $\mathcal{F}_V$. As stated in the theorem, we assume $\mathcal{T}_V \subseteq \mathcal{D}_V$, where node $u \in \mathcal{D}_V$ have degree $d$; proportion $h$ of their neighbors belong to the same class, while proportion $1-h$ of them belong to any other class uniformly. And for each node $v \in \mathcal{V}$ the node features are given as $x_v = p \cdot \text{onehot}(y_v) + \frac{1-p}{|Y|} \cdot 1$ for each node $v \in \mathcal{V}$. Then, after the first layer, we have $[A_1X]_{\mathcal{F}_V} = \frac{1}{|Y|}B((hd + 1)p, \frac{1-b}{|Y|-1})$, and after the second layer $[\tilde{A}_s^2X]_{\mathcal{F}_V} = \frac{1}{(|Y|-1)}B((h(d-1)|Y|-1)+b, \frac{1-b}{|Y|-1})$, where $S_1 = ((h|Y|-1)d + |Y|-1)^2p$, and $T_1 = ((h|Y|-1)d + |Y|-1)^2p$.

For $[Y]_{\mathcal{F}_V}$ and $[\tilde{A}_s^2X]_{\mathcal{F}_V}$, we can find the optimal weight matrix $W_*$ such that $[\tilde{A}_s^2X]_{\mathcal{F}_V}W_* = [Y]_{\mathcal{F}_V}$, making the cross-entropy loss $L([z]_{\mathcal{F}_V}, [Y]_{\mathcal{F}_V}) = 0$. To find $W_*$, we can proceed as follows. First, sample one element from each class to form a smaller set $\mathcal{T}_S \subset \mathcal{T}_V$. Therefore, we have $[Y]_{\mathcal{T}_S} = I[y|x,y]$ and $[\tilde{A}_s^2X]_{\mathcal{T}_S} = C^*(S_1,T_1)$. Note that $[\tilde{A}_s^2X]_{\mathcal{T}_S}$ is a circulant matrix, and therefore its inverse exists. Using the Sherman-Morrison formula, we have

$$([\tilde{A}_s^2X]_{\mathcal{T}_S})^{-1} = \frac{1}{p(d(h|Y|-1)+|Y|-1)^2}([Y]_{\mathcal{T}_S})^{-1} + \frac{1}{p(d(h|Y|-1)+|Y|-1)^2}([Y]_{\mathcal{T}_S})^{-1}.$$

Now, let $W_*$ be $([\tilde{A}_s^2X]_{\mathcal{T}_S})^{-1}$, then $[z]_{\mathcal{T}_S} = [\tilde{A}_s^2X]_{\mathcal{T}_S}W_* = [Y]_{\mathcal{T}_S}$, where $W_*$ satisfies $L([z]_{\mathcal{T}_S}, [Y]_{\mathcal{T}_S}) = 0$.

Attack loss under evasion attacks. Now consider an arbitrary target node $v \in \mathcal{D}_V$ with class label $y_v \in Y$, and a unit structural perturbation leveraging gambit node $u \in \mathcal{V}$ with degree $d_v$ that affects the predictions $z_v$ of node $v$ made by GNN $f^{(2)}_s$. Without loss of generality, we assume node $v$ has $y_v = 1$.

Proof for Thm. 1. We consider a unit perturbation in the direct (1-hop) neighborhood $N(v)$ of node $v$. For simplicity of derivation, we assume that the perturbation does not change the row-stochastic normalization of $A_v$, and only affects the aggregated feature vectors of the target node $v$. In the case of $\delta_1 = 1$ and $\delta_2 = 0$, we have $[A_1X]_{\mathcal{F}_V} - [A_1X]_{\mathcal{F}_V} = \delta_1 [d_v + 1] (1-p) \cdot \left(\frac{1}{|Y|} + p\cdot \frac{1}{|Y|}\right)$, and $[\tilde{A}_s^2X]_{\mathcal{F}_V} - [\tilde{A}_s^2X]_{\mathcal{F}_V} = \Delta [d_v + 1] (1-p) \cdot \left(\frac{1}{|Y|} + p\cdot \frac{1}{|Y|}\right)$, where $\Delta = d_v (p(d(h|Y|-1) + |Y|-2) + (d + 2)(|Y|-1)p + d + 2)$.

Writing $S_2 = d_v (p(d(h|Y|-1) + |Y|-2) + (d + 2)(|Y|-1)p + d + 2)$, we have $\Delta = d_v (p(d(h|Y|-1) + |Y|-2) + (d + 2)(|Y|-1)p + d + 2)$.

On the perturbed graph, the CM-type attack loss is calculated as $\Delta_{\text{atk}}(x_v) = -(x_v' + \max_{y \neq y_v} x_v'^y)$. Since $\Delta_{\text{atk}}(x_v) = 0$ on clean data, the change in attack loss before and after attack is

$$\Delta_{\text{atk}} = \Delta_{\text{atk}}(x_v) - \Delta_{\text{atk}}(x_v) = \frac{\delta_1}{(d_v + 1)^2} \cdot \left(\frac{1}{|Y|} + \frac{1}{|Y|}\right) \cdot \frac{1}{(d_v + 1)^2} \cdot \left(\frac{1}{|Y|} + \frac{1}{|Y|}\right).$$

Solving the system of inequalities for $\delta_1$ under constraints $\Delta_{\text{atk}} > 0$, $h \in [0,1]$, $|Y| \geq 2$ and $d_v$, $|Y| \in \mathbb{Z}^+$, we get the range of $\delta_1$ as

$$\begin{cases}
\delta_1 < 0, \text{ when } 0 < d < |Y| - 2 \\
\delta_1 < 0, \text{ when } d > |Y| - 2 \text{ and } d_v < d_1 \\
\delta_1 < 0, \text{ when } d > |Y| - 2 \text{ and } d_v \geq d_1 \text{ and } 1 \leq h > h_1,
\end{cases}$$

$$\delta_1 > 0, \text{ when } d > |Y| - 2 \text{ and } d_v \geq d_1 \text{ and } 0 \leq h < h_1.$$
where $d_1 = \frac{(d_2+1)(|Y|-1)}{d_1}$ and $h_1 = \frac{d_2(d_1 - |Y| + 1)(|Y|-1)}{d_1}$. Note that the above solution is not applicable when $h = \frac{d_2+1}{|Y|}$, in which case $d(h|Y|-1) + |Y|-1 = 0$ and the solution of optimal weight matrix $W_* = \left(\left[A^2 X|\cdot 0^1\right]_\delta\right)^{-1}$ is undefined.

In the case of $\delta_1 = 0$ and $\delta_2 = \pm 1$, following a similar derivation to that in the previous case, we can compute the change in the CM-type attack loss before and after attack as

$$\Delta L^C_M = \frac{\delta_2(d_1 + 1)(|Y|-1)S_1}{(d_1 + 1)^2(d(h|Y|-1) + |Y|-1)^2}.$$ (5)

Solving the same system of inequalities as in the previous case for $\delta_2$, we have $\delta_2 = -\delta_1$, where $\delta_1$ is bounded by Eq. (4). Note that the above solution is again not applicable when $h = \frac{d_2+1}{|Y|}$, and we always have $h_1 = \frac{1}{|Y|} = \frac{(d_2+1)(|Y|-1)}{d_1} < 0$ when $|Y| > 1$.

2. Unit perturbation in 2-hop neighborhood $N_2(v)$. We now consider a unit perturbation in the 2-hop neighborhood $N(v)$ of node $v$. In this case we will have $\left[A^2 X|\cdot 0^1\right]_\delta = \left[A^2 X|\cdot 0^1\right]_\delta$.

In the case of $\delta_1 = \pm 1$ and $\delta_2 = 0$, we have $\left[A^2 X|\cdot 0^1\right]_\delta = \left[A^2 X|\cdot 0^1\right]_\delta$.

where $2S_2 = \frac{d_1}{(d_1+1)(|Y|-1)} + (\frac{|Y|-1}{|Y|})p + 1$ and $T_2 = \frac{d_1}{(d_2+1)(|Y|-1)} + (\frac{|Y|-1}{|Y|})p + 1$.

By multiplying $\left[A^2 X|\cdot 0^1\right]_\delta$ with $W_*$, we can get the predictions $\hat{y}_2$ after perturbations. Following a similar derivation as before, we can get the change in the CM-type attack loss before and after attack as

$$\Delta L^C_M = \frac{(\delta_1(d_1 + 1)(|Y|-1) + (\frac{|Y|-1}{|Y|})d_2)}{(d_1 + 1)^2(d(h|Y|-1) + |Y|-1)^2}.$$ (6)

Solving for $\delta_1$ as in previous cases, we get the valid range of $\delta_1$ as

$$\delta_1 < 0, \text{ when } d_2 > |Y| - 1$$

$$\delta_1 = 0, \text{ when } d_2 \geq |Y| - 1 \text{ and } \frac{d_2^2 - |Y| + 1}{|Y|} < h \leq 1$$

$$\delta_1 > 0, \text{ when } d_2 > |Y| - 1 \text{ and } 0 \leq h < \frac{d_2^2 - |Y| + 1}{|Y|}.$$ (7)

Note that the above solution is not applicable when $h = \frac{d_2+1}{|Y|}$, in which case $d(h|Y|-1) + |Y|-1 = 0$ and the solution of optimal weight matrix $W_* = \left(\left[A^2 X|\cdot 0^1\right]_\delta\right)^{-1}$ is undefined.

For the case $\delta_1 = 0$ and $\delta_2 = \pm 1$, following a similar derivation to that in the previous case, we can compute the change in the CM-type attack loss before and after attack as

$$\Delta L^C_M = \frac{(d_1 + 1)(|Y|-1) + (\frac{|Y|-1}{|Y|})d_2}{(d_1 + 1)^2(d(h|Y|-1) + |Y|-1)^2}.$$ (8)

Solving the same system of inequalities as in the previous cases for $\delta_2$, we have $\delta_2 = -\delta_1$, where $\delta_1$ is bounded by Eq. (7). Note that the above solution is again not applicable when $h = \frac{d_2+1}{|Y|}$, and we always have $\frac{d_2^2 - |Y| + 1}{|Y|} = \frac{1}{|Y|} > 0$ when $|Y| < 2$.

Summary and validation of theorem statements. Based on our discussions, we validate our statements in the theorem next.

1. From Eq. (4), (7), we observe that for both direct attacks in 1-hop neighborhood $N(v)$ and indirect attacks in 2-hop neighborhood $N_2(v)$, when $h \geq \frac{1}{|Y|}$, the attack loss increases only if $\delta_1 < 0$ (i.e., removal of a homophilous edge or path to node $v$), or if $\delta_2 > 0$ (i.e., addition of a heterophilous edge or path to node $v$).

2. From Eq. (3), (5), (6) and (8), we can show that $\Delta L^C_M > \Delta L^C_M > \Delta L^C_M > 0$ for a unit perturbation increasing loss $\Delta L^C_M$.

Proof for Thm. 2. The theorem statements can be directly validated from Eq. (4) and its discussions in Proof for Thm. 1, which solves $\delta_1$ and $\delta_2$ for an effective unit perturbation in direct neighborhood $N(v)$ of node $v$. We similarly note that the conclusions are not applicable when $h = \frac{d_2+1}{|Y|}$, in which case the solution of optimal weight matrix $W_*$ is undefined.

Proof for Thm. 3. Here, we mainly focus on analyzing $\Delta L^C_M$ for the layer defined as $f(A, X; \alpha) = (1 - \alpha)A + \alpha X$, as the layer defined as $f_2(A, X) = A_0X$ is a special case when $\alpha = \frac{1}{1+\alpha}$. We follow a similar process as in Proof of Thm. 1.

Layer $f(A, X; \alpha) = (1 - \alpha)A + \alpha X$.

We first derive the optimal weight matrix $W_*$ in a stylized learning setup as in Proof A. Following a similar process, for this GNN layer we have

$$\langle 1 - \alpha \rangle A + \alpha X \rangle \right|_{\delta_2} = \frac{1}{2} + C \left( \frac{\alpha + h - \alpha h}{|Y| + 1} \right)$$

and $W_* = \frac{\alpha}{2} (\alpha h - \alpha h + 1) |Y| + C \left( \frac{\alpha + h - \alpha h}{|Y| + 1} \right)$. (10)

Now as in Proof for Thm. 1, we consider an arbitrary target node $v \in D_Y$, and a unit structural perturbation leveraging gambit node $u \in V$ with degree $d_u$ that affects the predictions $\hat{y}_2$ of node $v$ made by layer $(A, X, \alpha)$. Note that we only discuss the case of direct structural perturbation to the 1-hop neighborhood $N(v)$ of target node, as indirect perturbations do not affect the predictions $\hat{y}_2$ for node $v$ produced by a single GNN layer. Following a similar derivation as in Proof for Thm. 1, in the case of $\delta_1 = \pm 1$ and $\delta_2 = 0$, or $\delta_1 = 0$ and $\delta_2 = \pm 1$, the change in the CM-type attack loss $\Delta L^C_M$ for GNN layer $(A, \alpha)$ considering both $\delta_1$ and $\delta_2$ is

$$\Delta L^C_M = \frac{\alpha}{2} (\alpha h - \alpha h + 1) |Y| + C \left( \frac{\alpha + h - \alpha h}{|Y| + 1} \right)$$

and $W_* = \frac{\alpha}{2} (\alpha h - \alpha h + 1) |Y| + C \left( \frac{\alpha + h - \alpha h}{|Y| + 1} \right)$. (10)

Layer $f_2(A, X) = A_0X$. This formulation is a special case of the previously discussed $f(A, X; \alpha)$ formulation when $\alpha = \frac{1}{1+\alpha}$. We denote the change in the attack loss for this layer as $\Delta L^C_M$.

Comparison of increase in attack loss $\Delta L^C_M$. Solving the following system of inequalities for variable $\alpha$ under constraints $\Delta L^C_M > 0$, $h \in [0, 1]$, $\alpha \geq 2$, $d_u |Y| \in \mathbb{Z}^*$ and $\delta_1, \delta_2 \in \{0, 1, 0\}$, we get the valid range of $\alpha$ as

$$\alpha > \frac{1}{|Y|} \text{ or } \alpha < \frac{1}{|Y|}$$

From Eq. (10), we observe that when $h > \frac{1}{|Y|}$, a unit perturbation increasing $L^C_M$ as discussed in Thm. 1 (i.e. $\delta_1 = 1$ and $\delta_2 = 0$), or $\delta_1 = 0$ and $\delta_2 = 1$), which satisfies the condition $\delta_1 < \delta_2$, will thus lead to a strictly smaller increase in $L^C_M$ layer $f(A, X; \alpha)$ than the increase $\Delta L^C_M$ for layer $f_2(A, X)$ if $\alpha > \frac{1}{|Y|}$.