# Disparate Vulnerability in Link Inference Attacks against Graph Neural Networks

Da Zhong
Stevens Institute of Technology
Hoboken, NJ, USA
dzhong2@stevens.edu

Ruotong Yu
University of Utah
Salt Lake City, UT, USA
routoy@cs.utah.edu

Kun Wu
Stevens Institute of Technology
Hoboken, NJ, USA
kwu14@stevens.edu

Xiuling Wang
Stevens Institute of Technology
Hoboken, NJ, USA
xwang193@stevens.edu

Jun Xu
University of Utah
Salt Lake City, UT, USA
junxzm@cs.utah.edu

Wendy Hui Wang
Stevens Institute of Technology
Hoboken, NJ, USA
hwang4@stevens.edu

## ABSTRACT

Graph Neural Networks (GNNs) have been widely used in various graph-based applications. Recent studies have shown that GNNs are vulnerable to link-level membership inference attacks (LMIA) which can infer whether a given link was included in the training graph of a GNN model. While most of the studies focus on the privacy vulnerability of the links in the entire graph, none have inspected the privacy risk of specific subgroups of links (e.g., links between LGBT users). In this paper, we present the first study of disparity in subgroup vulnerability (DSV) of GNNs against LMIA. First, with extensive empirical evaluation, we demonstrate the existence of non-negligible DSV under various settings of GNN models and input graphs. Second, by both statistical and causal analysis, we identify *the difference between three specific graph structural properties of subgroups* as one of the underlying reasons for DSV. Among the three properties, the difference between subgroup density has the largest causal effect on DSV. Third, inspired by the causal analysis, we design a new defense mechanism named FairDefense to mitigate DSV while providing protection against LMIA. At a high level, at each iteration of target model training, FairDefense randomizes the membership of edges in the training graph with a given probability, aiming to reduce the gap between the density of different subgroups for DSV mitigation. Our empirical results demonstrate that FairDefense outperforms the existing defense methods in the trade-off between defense and target model accuracy. More importantly, it offers better DSV mitigation.

## KEYWORDS

Membership inference attacks, Graph Neural Networks, fair privacy.

## 1 INTRODUCTION

Recently, there has been increasing interest in designing deep learning approaches for graph analysis, resulting in the creation of graph neural networks (GNNs) [30, 60]. GNNs have been demonstrated to be powerful in modeling graph-structured data and have become a widely applied method for large-scale graph analysis.

Despite their effectiveness in a variety of analytic tasks, GNNs, like other neural models, are prone to privacy attacks. Recent studies [34, 35] have shown that GNNs are vulnerable to several privacy attacks, such as model extraction attacks [61, 75], property inference attacks [26, 76], and membership inference attacks [34, 35, 54]. In this paper, we focus on the *link-level membership inference attack* (LMIA) [34, 72], whose goal is to infer whether a particular link (edge) exists in the training graph of the target GNN model. As the links in the training graph may represent critical private information such as political or religious affiliation [2, 29], the privacy vulnerability of GNNs against LMIA raises serious concerns.

Recent studies [34, 72] have shown that LMIA can recover a significant amount of private edges from the training graph effectively. While these studies mainly focus on the privacy leakage across all the links in the entire graph, *whether such privacy leakage is evenly distributed across the links that connect different types of nodes* remains largely unexplored. Therefore, in this paper, we aim to investigate the disparate vulnerability of GNNs against LMIAs at the link level, i.e., whether the links between the users in some specific demographic groups (e.g., the minority group) are more vulnerable to LMIAs than those between users in other groups.

The conventional fairness research [8, 32] defines fairness across various population groups (e.g., males and females). While we mainly consider the link-level fairness in terms of privacy disparity, the link-level disparate vulnerability can lead to the node-level disparate vulnerability. Consider an example of a social network graph whose node features include users' sexual orientation. If the social connections (links) between lesbian, gay, bisexual, and transgender (LGBT) users are easier to be attacked than the links between non-LGBT users, LGBT users will suffer from higher privacy threats than non-LGBT users. For example, it will be more difficult for LGBT users to hide their sexual orientation than non-LGBT ones as the sexual orientation information can be inferred from their (predicted) friends in the social network.

Recently, the disparity in privacy vulnerability of membership inference attacks has been investigated for the learning models over non-graph data [11, 44, 78]. Through these investigations, some inherent data characteristics such as *imbalanced subgroup size* have been identified as the underlying reasons for the disparate vulnerability [11, 44]. For example, the minority group (i.e., the group


*Proceedings on Privacy Enhancing Technologies YYYY(X), 1–21*

https://doi.org/XXXXXXX.XXXXXXX

of smaller size) always has higher privacy membership leakage than the majority group. However, our analysis in Section 5 will unveil that the disparity in subgroup vulnerability of GNNs against LMIA is no longer attributed to these characteristics. For example, both minority and majority link subgroups witness similar amounts of privacy leakage to LMIA. This raises the critical need for a new investigation into the disparity in privacy vulnerability of membership inference attacks under the graph learning setting.

This paper presents the first study of the *disparity in subgroup vulnerability* (DSV) of GNNs against LMIA. We aim to answer the following research questions:

- **RQ1**: *Does the disparity in subgroup vulnerability exist for the representative GNN models (i.e., is there a link subgroup always more vulnerable to LMIA than the others)?*
- **RQ2**: *Does any graph structural property cause the vulnerability disparity?*
- **RQ3**: *How to mitigate the disparity in subgroup vulnerability and provide fair privacy protection against LMIA across all subgroups?*

To answer these three questions, we make the following contributions:

► **Empirically measuring disparity in subgroup vulnerability.** We perform an extensive set of experiments on two representative GNNs (Graph Convolution Network (GCN) [42] and Graph Attention Network (GAT) [68]) and three real-world social network graphs. We partition the edges in the social network graphs into different subgroups based on whether the edges connect users with similar values of the protected attribute (e.g., gender and education level). By quantifying DSV as the gap between the attack performance of different link subgroups, where the attack performance is evaluated by two different metrics (Balanced Attack Accuracy (BAA) [9] and F1-score), our empirical evaluation unveils the existence of non-negligible DSV of a state-of-the-art LMIA attack [34] in all the settings we examined. For example, in the Spammer social graph[1], the links between male users (M-M subgroup) are much more vulnerable to LMIA (with BAA as 0.79) than the links between female users (F-F subgroup), whose BAA is as low as 0.56. This causes a significant DSV between these two subgroups.

► **Unveiling the underlying reasons behind DSV.** To understand why DSV exists, we analyze how various graph structural properties of different subgroups affect the LMIA performance of these subgroups, and identify a strong Pearson correlation between LMIA performance and three structural properties: *density*, *average node similarity*, and *average edge betweenness centrality*. For example, we identify a strong negative correlation between group density and attack performance. Following the aforementioned observation of DSV in the Spammer dataset, as the M-M subgroup in the Spammer social graph is the sparsest among the three subgroups (M-M, F-M, and F-F), the edges in the M-M subgroup are more vulnerable to LMIA than those in the other two subgroups. We investigate the reason behind this correlation and find that the connected node pairs (members) in a subgroup of a lower density are more distinguishable from the disconnected ones (non-members) than those in a subgroup of a higher density.

We further perform extensive counterfactual causal analysis of the three structural properties and discover that the difference between these properties of subgroups has a strong causal effect on DSV. In other words, one main reason that DSV exists between different subgroups is that these subgroups have different densities, different node similarities, and different edge betweenness centrality. In particular, the difference between the density of subgroups has the largest causal effect on DSV.

► **Designing defense mechanisms to mitigate DSV.** Based on our analysis of the underlying causes, we design a new algorithm named FairDefense which can mitigate DSV while providing protection against LMIA. Essentially, FairDefense perturbs the structure of the training graph during the training of the target model by randomizing the membership of edges with some given probability at each iteration of training. To minimize the impact of perturbation on target model accuracy, instead of continuously perturbing the graph that has been noised in the previous iterations, FairDefense always applies randomization on the original graph, so that the perturbation in the previous iterations is abandoned and will not be accumulated during training. We formally prove that FairDefense guarantees to reduce the gap between the density of different subgroups and thus achieves DSV mitigation. Our empirical results show that FairDefense outperforms four baseline methods in both defense effectiveness and DSV mitigation. Furthermore, FairDefense has a better trade-off between defense effectiveness and target model accuracy than the baseline methods.

## 2 GRAPH NEURAL NETWORKS

Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the links (edges) between the nodes, the goal of a graph neural network (GNN) is to utilize both graph structure and node features in $\mathcal{G}$ to learn a low-dimension representation (or embedding, denoted as $h$) of each node. Node embedding can be used for various downstream tasks such as link prediction and node classification. In this paper, we consider social network graphs where the nodes represent users, and the links represent users' social relationships. The nodes are associated with features that capture users' personal information (e.g., age and gender).

In this paper, we consider *message-passing GNN*, one of the most popular GNN models. Specifically, the message exchange is defined by a *message function*, whose input is a node's feature and output is a message, and an *aggregation function*, whose input is a set of messages and output is the updated node feature. At each round of message exchange, each node sends messages to its neighbors and aggregates incoming messages from its neighbors. Each message is transformed at each link through a message function $MSG()$, and then they are aggregated at each node via an aggregation function $AGG()$. In this paper, we consider the transductive setting (i.e., all nodes in the graph are available at training time).

**Graph convolution network (GCN).** GCN [42], one of the widely-used GNN models, follows the message-passing framework. For the $k$-th iteration of message-passing (i.e., the $k$-th layer of a

[1]More details of the Spammer social graph are provided in Section 3

GCN), its message function $MSG()$ is defined as follows:

$$H^k = \sigma(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}H^{(k-1)}W^{k-1}) \quad (1)$$

where $H^k$ is the node embedding at the $k$-th layer, $\tilde{A} = A + I$ denotes the adjacency matrix $A$ of the given graph $\mathcal{G}$ added with self-connection, $I$ is the identity matrix, $D$ is a diagonal matrix that $D_{ii} = \sum_j A_{ij}$, $W^k$ denotes the weight matrix at the $k$-th layer, and $\sigma(.)$ denotes an activation function such as ReLU. The initial $H^0$ can be set as the matrix of node feature vectors.

The $AGG()$ function of GCN computes the embedding $h_i$ of the node $v_i$ at the $k$-th iteration as follows:

$$h_i^k = \sigma(\sum_{j \in N_i} c_{ij} W^k h_j^{k-1}), \quad (2)$$

where $c_{ij} = \frac{1}{\sqrt{|N_i||N_j|}}$, $N_i$ ($N_j$) is the number of neighbors of node $v_i$ ( $v_j$), and $W^k$ denotes the weight matrix at the $k$-th layer.

**Graph attention network (GAT).** GAT [68] uses the same messaging function (Eqn. (1)) as GCN. The key difference between GAT and GCN is the aggregation function $AGG()$. While GCN considers the coefficient $\frac{1}{\sqrt{|N_i||N_j}}$ to indicate the weight of the link $(v_i, v_j)$ based on the local graph structure, GAT utilizes the *attention mechanism* [4] to compute the link weights by attending over the hidden features of neighbors so that links with more important connecting neighbors have higher weights. Formally, the attention coefficient $a_{ij}$ is computed from node features, which is then passed into an attention function for the update. We omit the details of the attention function as they are irrelevant to the following discussions. Then the $AGG()$ function of GAT is formed as:

$$h_i^k = \sigma(\sum_{j \in N_i} a_{ij} W^k h_j^{k-1}). \quad (3)$$

In this paper, we target both GCN and GAT. We consider node classification as the learning task. Specifically, the GNN model outputs the node classification results in the format of the probability distribution over a set of possible labels, abbreviated as *posteriors*.

# 3 LINK MEMBERSHIP INFERENCE ATTACKS AGAINST GNNS

In this section, we first present the details of the link membership inference attacks that we consider in the paper. Then we present the empirical evaluation of the performance of the attacks.

## 3.1 Attack Details

In general, membership inference attacks (MIA) aim to infer whether a data point is included in the training data of the target model [64]. Recently, MIA has been adapted to GNNs. Specific *node-level* [35] and *link-level* MIA attacks [34] are designed to predict the existence of particular nodes and links in the training graph, respectively. In this paper, we consider the *link-level* MIA (referred to as LMIA for simplicity) [34] as the attack model. We do not consider the node-level MIA due to its triviality for both GCN and GAT models where each node is necessarily included in the training data. Next, we briefly discuss the details of the attack presented in [34]. Note that our focus is not on the design of a new LMIA attack. Instead, we aim to investigate the *disparity in privacy vulnerabilities* by the existing LMIA attack [34].

**Adversary knowledge.** The LMIA in [34] considers two types of adversary knowledge: *partial graph* (i.e., a subset of links from the training graph) and *node features*. Based on such adversary knowledge, [34] develops eight types of attacks. Out of these eight attacks, we focus on two of them, **Attack-3** and **Attack-6**, that deliver the highest attack performance. In the rest of this paper, we refer to these two attacks as **Attack A** and **Attack B**. Attack A assumes that the adversary has access to a partial graph, while Attack B assumes that the adversary can access both partial graph and node features in the training graph. These two types of adversarial knowledge are easily accessible in real-world settings. For example, many users on social network platforms configure their privacy settings as open to the public. Such a setting makes their friend lists (i.e., the partial graph) and user profiles (i.e., node features) available to the adversary. Note that both attacks do not use *shadow graphs* for the training of the attack model. Instead, they use the links from the known partial graph as the ground truth label to train the attack model.

**Design of Attacks A and B.** At a high level, the LMIA attack model is a binary classifier that is trained to distinguish a GNN model's behavior on its training members from that on the non-members. Consider a GNN model $f$ whose learning task is node classification as the target model. The output of the GNN model for a given node $u$ (denoted as $f(u)$) is a vector of posterior probability, with the $i$-th probability in $f(u)$ indicating the likelihood that the node belongs to the $i$-th class. An important property of this type of GNN models is that if two nodes are connected (i.e., on a member edge), their posterior outputs should be more similar than the outputs if they are disconnected (i.e., on a non-member edge). This property is utilized by both Attacks A and B [34]. Table 1 presents the features of both attacks.

For Attack A, the adversary measures the distance between the posterior outputs of any two nodes. Details of the distance metric $d()$ can be found in [34]. In addition, the adversary considers a pairwise vector operation $\circ$, and applies it to both the posterior outputs of any two nodes and the entropy $e$ of the posterior outputs. Finally, the adversary extracts the features of Attack A from the distance $d(f(u), f(v))$, the output of pairwise vector operation on the posterior outputs $\circ(f(u), f(v))$, and the output of the vector operations on the entropy of the posterior outputs $\circ(e(f(u)), e(f(v)))$.

Compared with Attack A, Attack B has additional knowledge of node features. The adversary utilizes such additional knowledge to train another *reference model* $g$, which is an MLP model over node features only, with the intuition that the distance between the posterior outputs of the two connected nodes from the target model should be smaller than the corresponding distance from the reference model. Thus, besides the same LMIA features used by Attack A, the LMIA features of Attack B include the ones extracted from the outputs of the reference model $g$ and node features.

## 3.2 Attack Performance

In this subsection, we present the performance of the two attacks, aiming to justify that the attacks are ready for the analysis of DSV. All the experiments are performed on a server with eight NVIDIA A100 GPUs (40G). All the algorithms are implemented in Python

**Table 1: Features of Attacks A & B. $u, v$: two nodes; $f$: target model; $d$: distance function; $\circ$: pairwise vector operator; $e$: entropy; $g$: reference model; $A_u$: node feature of $u$.**

| Attack | $d(f(u), f(v))$ | $\circ(f(u), f(v))$ | $\circ(e(f(u)), e(f(v)))$ | $d(g(u), g(v))$ | $\circ(g(u), g(v))$ | $\circ(e(g(u)), e(g(v)))$ | $d(A_u, A_v)$ | $\circ(A_u, A_v)$ |
|---|---|---|---|---|---|---|---|---|
| Attack A | ✓ | ✓ | ✓ | × | × | × | × | × |
| Attack B | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Table 2: Description of the datasets.**

| | Facebook | Pokec | Spammer |
|---|---|---|---|
| # of node features | 574 | 275 | 3 |
| # of label classes | 4 | 2 | 2 |
| # of nodes | 1,034 | 4,037 | 10,000 |
| # of edges | 27,380 | 8,203 | 25,302 |

along with PyTorch. Each experiment is repeated 10 times and the average results are reported.

*3.2.1 Experimental Setup.* We describe datasets, target model setup, and performance metrics in this subsection.

**Datasets.** We use three real-world social network graphs: **Facebook** [45], **Pokec** [45], and **Spammer** [25], for target model training. Table 2 summarizes the main characteristics of the three social network graphs. More details of the three graphs are included in Appendix A.1.

**Target model and learning task.** We consider two representative GNN models, Graph Convolution Network (GCN) [42] and Graph Attention Network (GAT) [68], as the target models. We consider node classification as the learning task of the two models. The details of the setup (number of layers, learning rate, etc.) of the two GNN models are summarized in Appendix A.2.

**Training and testing of LMIA classifier.** Following the setting of [34][2], we randomly select 20% of the edges (*members*) and the same amount of originally non-existing edges (*non-members*) from the training graph to construct LMIA training data. The LMIA testing dataset consists of the remaining 80% member edges and the same amount of non-member edges which are randomly picked from the training graph (without overlap with those used for training). Besides the equal number of members and non-members in the whole data, we ensure each link group has the same number of member and non-member links.

**Evaluation metrics.** Regarding the target model, we measure its performance as its accuracy of node classification, i.e., the percentage of nodes that are classified correctly. In terms of the attack model, we consider two metrics to evaluate its performance, namely *balanced attack accuracy* and *F1-score*, which are popularly used in the literature on membership inference attacks:

- **Balanced attack accuracy (BAA)** [14, 33, 52, 64] is a standard accuracy metric that measures how often an attack correctly predicts membership (predicting members as member and non-members as non-member) on a balanced dataset of members and non-members [9]. Intuitively, larger BAA values indicate higher effectiveness of the attack.

[2]https://github.com/xinleihe/link_stealing_attack

**Table 3: Performance of target model.**

| Dataset | GAT | | GCN | |
|---|---|---|---|---|
| | Training | Testing | Training | Testing |
| Facebook | 82.40% | 81.16% | 85.99% | 85.65% |
| Pokec | 65.70% | 61.90% | 83.07% | 61.90% |
| Spammer | 69.83% | 68.10% | 70.26% | 71.33% |

**Table 4: Performance of LMIA Attack.**

| Dataset | Attack | GAT | | GCN | |
|---|---|---|---|---|---|
| | | BAA | F1 | BAA | F1 |
| Facebook | Attack A | 0.734 | 0.769 | 0.810 | 0.778 |
| | Attack B | 0.748 | 0.778 | 0.816 | 0.813 |
| Pokec | Attack A | 0.683 | 0.690 | 0.627 | 0.716 |
| | Attack B | 0.721 | 0.707 | 0.717 | 0.776 |
| Spammer | Attack A | 0.687 | 0.731 | 0.641 | 0.669 |
| | Attack B | 0.684 | 0.723 | 0.589 | 0.629 |

- **Attack F1-score (F1)** [36, 38, 56] combines the precision (*pre*) and recall (*Rel*) of the attack results into one value. Informally, *Pre* measures the proportion of the inferred members that are indeed members, and *Rel* measures the proportion of the real members that are correctly inferred by the attack. Based on the definition of precision and recall, the F1-score of the attack is measured as: $F1 = \frac{2*Pre*Rel}{Pre+Rel}$. Intuitively, a higher F1-score indicates that the attack is more effective in link inference.

Besides these two metrics, some new metrics (e.g., *true-positive rate (TPR) at low false-positive rates (FPR)* [10]) have been proposed recently for LMIA evaluation. However, we found that the metric of TPR@FPR is not appropriate for the measurement of DSV. More detailed explanations can be found in Section 4.2.

*3.2.2 Attack Evaluation.* Before we evaluate the attack performance, we evaluate the performance of the target model. Table 3 presents the target model performance of both GCN and GAT models on three datasets. Overall, both models achieve acceptable classification performance, which is significantly higher than random guess. Furthermore, both models do not show noticeable over-fitting except for one single setting (GCN on Pokec dataset).

Table 4 presents the performance of both Attacks A and B (both BAA and F1 metrics). Overall, LMIA presents high effectiveness in all the settings, evidently outperforming the random guess (0.5 for both BAA and F1-score). We also observe that the performance of Attack B is better than Attack A in most of the settings, thanks to the additional adversary knowledge of node features. The only exception is the Spammer dataset, where the performance of Attack

A is slightly better than Attack B. Our explanation of this exception can be found in Appendix B.1.

# 4 DISPARATE SUBGROUP VULNERABILITY

Our empirical results in Section 3 show that the LMIA attacks are effective across the whole graph. However, whether the attack performance varies across different link subgroups remains unclear. Disparate vulnerability in LMIA can raise concerns if some subgroups, particularly those serve the minority population (e.g., LGBT population and their social connections) are more vulnerable than the other subgroups. In this section, we study the issue of disparate vulnerability in LMIA against GNNs. First, we formally define disparity in subgroup vulnerability. Next, we empirically evaluate the disparity in subgroup vulnerability.

## 4.1 Conventional Group Fairness in Machine Learning

Most existing notions of group fairness in machine learning (ML) systems require certain statistic of interest to be approximately equalized across groups [8, 32]. In general, they apply to a dataset of domain $S \times X \times Y$, where $S$, $X$ and $Y$ respectively denote the *protected attributes* (e.g., gender), the non-protected attributes, and the outcome feature. A fair classification model $M$, as per those notions, should not present discriminatory favor/disfavor towards a particular group defined by the protected attribute (e.g., female defined by gender).

Mathematically, the fairness notions have various forms. In this paper, we adapt the one termed *accuracy parity* [6, 7, 13] to our problem setting. The rationale is that accuracy parity properly reflects the inconsistent behaviors of LMIA against different demographic groups. Formally,

Definition 1 (Accuracy Parity [6, 7, 13]). *Given an ML model $\mathcal{T}$, a pre-defined metric ACC that measures the accuracy of the prediction output by $\mathcal{T}$, and two groups $G_i$ and $G_j$, then $\mathcal{T}$ satisfies* ***accuracy parity*** *if $ACC(G_i) = ACC(G_j)$.*

The violation of accuracy parity is known as *disparate mistreatment* [74]. The extent of disparate mistreatment can be measured by *accuracy gap*:

Definition 2 (Accuracy gap). *Following the notations in Definition 1, the* accuracy gap *of the ML model $\mathcal{T}$ on $G_i$ and $G_j$ is $\Delta = |ACC(G_i) - ACC(G_j)|$.*

Intuitively, $\Delta = 0$ indicates that $\mathcal{T}$ satisfies accuracy parity.

## 4.2 Formal Definition of Disparity in Subgroup Vulnerability

Our definition of disparity in subgroup vulnerability (DSV) is adapted from the conventional group fairness in machine learning. Unlike conventional data, the grouping of graph data can be defined in two ways, including *node subgroups* and *link subgroups*.

**Node subgroups.** Intuitively, node subgroups refer to the nodes grouped by specific node features (e.g., race, gender, ethnicity, etc.). Formally, consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ has a set of features $\mathcal{F}$ describing the individual's information. These features are split into two types: *protected node features $S$* and *non-protected node features $X$*. The protected node features are equivalent to the protected attributes in conventional fairness literature. Common protected node features are those carrying demographic properties such as race and gender that are mostly witnessed in social network graphs. To better bound our research, we assume the well-defined protected node features are available, and only one protected node feature is present. Based on the protected node feature, the nodes can be partitioned into different subgroups called *node subgroups*. For example, when considering *gender* as the protected feature, the nodes can be partitioned into two subgroups: males and females.

**Link subgroups.** Based on the definition of node subgroups, links can be grouped by the node features of their connected nodes. Formally, a *link subgroup $G$* is defined by the values associated with the protected node feature $S$ in $G$: $G = \{e(u, v) | u.S = s_u, v.S = s_v\}$. We denote the link subgroup $G$ as a *($s_u$-$s_v$) group* for simplicity. For example, when gender is used as the protected node feature, there are two node subgroups (Male and Female) and three link subgroups {*Male-Male (M-M), Female-Female (F-F), Male-Female (M-F)*}. In general, given a protected node feature that has $k$ distinct values (and thus $k$ node groups), there are $\frac{(k+1)k}{2}$ link subgroups.

**Subgroup vulnerability.** As we consider two evaluation metrics (BAA and F1-score) for the attacks, we adapt both evaluation metrics to the group-level measurement. Formally, give a link subgroup $G$, its *Balanced Attack Accuracy* $BAA(G)$ is measured as the percentage of edges in $G$ that are correctly classified (as either member or non-member):

$$BAA(G) = \frac{\sum_{e \in G} \mathbb{1}(\mathcal{M}(\mathcal{T}(e)) = b(e))}{|G|}, \tag{4}$$

where $b(e)$ indicates the ground truth membership of the edge $e$, and $\mathbb{1}() = 1$ (0, resp.) if the membership prediction is correct.

Similarly, give a link subgroup $G$, its *F1-score* $F1(G)$ is measured as follows:

$$F1(G) = \frac{2 * Pre(G) * Rel(G)}{Pre(G) + Rel(G)}, \tag{5}$$

where $Pre(G)$ measures the proportion of the inferred members that are indeed members in $G$, and $Rel$ measures the proportion of the real members in $G$ that are correctly inferred by the attack.

**Disparity in subgroup vulnerability.** To quantify the varying performance of the attack on different link subgroups, we adapt the definition of accuracy gap (Def. 2) to LMIA setting to measure the disparity in subgroup vulnerability (DSV).

Intuitively, DSV measures the difference between the attack performance of each group. Formally, for any two subgroups $G_i$ and $G_j$, the *disparity in subgroup vulnerability (DSV)* is measured as follows:

$$DSV(G_i, G_j) = |PL(G_i) - PL(G_j)|, \tag{6}$$

where $PL()$ is a metric that evaluates the performance of the attack on $G$. In our settings, $PL()$ will be either BAA (Eqn. (4)) or F1-score (Eqn. (5)). According to the definition, $DSV(G_i, G_j) = 0$ indicates that there is no vulnerability disparity between $G_i$ and $G_j$.

Given a graph $\mathcal{G}$ that contains $t > 2$ link subgroups, there will be $L = \frac{t(t-1)}{2}$ pairs of these subgroups, each pair having a DSV value. To aggregate all these DSV values into one single value, we

**Table 5: Setup of node/link subgroups of three graphs.**

| Datasets | Protected node feature | Node subgroups | Link subgroups |
|---|---|---|---|
| Facebook | Gender | Female (F), Male (M) | F-F, F-M, M-M |
| | Education Level | College (C), Non-college (NC) | C-C, NC-C, NC-NC |
| Pokec | Gender | Female (F), Male (M) | F-F, F-M, M-M |
| | Marital status | Married (M), Not-Married (NM) | M-M, NM-M, NM-NM |
| Spammer | Gender | Female (F), Male (M) | F-F, F-M, M-M |

measure the DSV of graph $\mathcal{G}$ as the average of the DSV values of all subgroup pairs. Formally,

$$DSV(\mathcal{G}) = \frac{\sum_{\forall G_i, G_j \in \mathcal{G}} DSV(G_i, G_j)}{L}. \quad (7)$$

To this end, larger (smaller) DSV indicates a higher (lower) disparity in LMIA performance across different link subgroups.

**Inappropriateness of TPR@FPR metric [10] for DSV measurement.** Intuitively, the metric of *true-positive rate (TPR) at low false-positive rates (FPR)* (TPR@FPR) [10] can be adapted to the evaluation of DSV by measuring DSV as the gap between TPR of different subgroups. However, such adaption is incorrect. Let us explain the reasons. The evaluation of TPR@FPR requires a "universal" threshold value to determine FPR of the whole graph. However, as the subgroups have different attack performances, using a single threshold for all the subgroups will lead to different FPRs of the subgroups. Then measuring DSV as the difference between TPR of subgroups that can have different FPRs (e.g., TPR@5%FPR - TPR@10%FPR) is not fair. Indeed, as TPR@FPR considers the evaluation under the worst-case methodology, and different subgroups have different worst-case performance, we believe a metric that evaluates the average-case performance (e.g., BAA and F1-score) is more suitable than the TPR@FPR metric for DSV measurement.

## 4.3 Measuring Disparity in Subgroup Vulnerability

To answer question $RQ_1$ — *whether the representative GNN models have disparate vulnerabilities for different link subgroups*, we perform an extensive set of empirical studies to evaluate the amounts of DSV on both GCN and GAT. We use the same experimental settings as described in Section 3.2.1.

**Setup of node and link subgroups.** For each graph, we pick a node feature as the protected node feature and construct the corresponding node and link subgroups. For example, when *gender* is used as the node feature, we categorize the links into three subgroups: *Male-Male group* (M-M), *Female-Female group* (F-F), and *Male-Female group* (M-F). Table 5 shows the setup details of the node and link subgroups of the three graph datasets.

**Finding #1: non-negligible DSV exists.** Table 6 reports the attack performance of link subgroups for three graph datasets, with *Gender* as the protected node feature. The results of other protected node features can be found in Appendix B.2. Our main observation is that the attack has different performance across different link subgroups in all the settings. For example, as shown in Table 6, when launching Attack B against GCN model on Spammer dataset, the BAA performance can reach 0.786 on the subgroup $G_0$ (M-M group), but drops to 0.561 on $G_2$ (F-F group), leading to DSV as large as 0.225 between these two groups. In other words, the links between males are much more vulnerable to LMIA than those between females. Similar observation also holds on F1-score metric. For example, when launching Attack A against GAT model on Spammer dataset, the F1-score can reach 0.774 on $G_0$, but drops to 0.589 on $G_2$, leading to DSV as 0.19. We will explain the reason behind this finding in §5.

**Finding #2: Some subgroups are consistently more vulnerable than the others.** Our second finding is that some subgroups are always more vulnerable than the others regardless of the attack evaluation metric, the attack type, and the type of target GNN model. For example, the subgroup $G_0$, which is the one of the lowest density[3], always has the highest attack performance (BAA or F1-score), while the subgroup $G_2$, which is of the highest density, always has the lowest attack performance. This implies that graph structure plays an important role in the existence of DSV—some subgroups are inherently more vulnerable to the attacks than the others due to their graph structure.

# 5 UNDERLYING REASONS FOR DISPARITY IN SUBGROUP VULNERABILITY

In this section, we answer the research question $RQ_2$: *What are the causes of the disparity of subgroup vulnerability?* As **Finding #2** has shown that DSV can exist regardless of both target model and attack models, we mainly focus on the graph structural properties of link groups, aiming to find out which structural properties are the underlying factors of DSV. In particular, we measure the statistical correlation between the attack performance and the particular structural properties of each subgroup. Next, we first explain the graph structural properties that we examine (Section 5.1). Then we present our findings on the relationship between the examined structural properties and DSV (Section 5.2).

## 5.1 Graph Structural Properties under Investigation

Why does DSV exist? Prior works [19] have shown that one of the sources of the "unfairness" of GNNs is data bias. Can data bias lead to DSV too? To answer this question, we consider four types of graph structural properties that have been popularly considered for the research of algorithmic fairness in graph learning: *group size* [20, 49], *group density* [39, 77], *average node similarity* [18, 55],

[3]More details of subgroup density will be explained in Section 5

**Table 6: Disparity in subgroup vulnerability (Gender as the protected node feature). $G_0$, $G_1$ and $G_2$ are link subgroups sorted by the group density in ascending order (Facebook & Pokec datasets: $G_0$ (F-F), $G_1$ (F-M), $G_2$ (M-M); Spammer dataset: $G_0$ (M-M), $G_1$ (F-M), $G_2$ (F-F)). The subgroups of the highest and lowest attack performance are marked with green and pink, respectively.**

**(a) Balanced attack accuracy (BAA)**

| Dataset | Attack | GAT | | | | GCN | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $G_0$ | $G_1$ | $G_2$ | DSV | $G_0$ | $G_1$ | $G_2$ | DSV |
| Facebook | Attack A | 0.804 | 0.789 | 0.680 | 0.083 | 0.888 | 0.875 | 0.748 | 0.093 |
| | Attack B | 0.808 | 0.804 | 0.693 | 0.077 | 0.891 | 0.881 | 0.754 | 0.091 |
| Pokec | Attack A | 0.714 | 0.711 | 0.618 | 0.064 | 0.688 | 0.635 | 0.578 | 0.073 |
| | Attack B | 0.800 | 0.751 | 0.673 | 0.085 | 0.784 | 0.757 | 0.671 | 0.075 |
| Spammer | Attack A | 0.794 | 0.673 | 0.583 | 0.139 | 0.786 | 0.613 | 0.561 | 0.150 |
| | Attack B | 0.760 | 0.674 | 0.609 | 0.101 | 0.644 | 0.580 | 0.541 | 0.069 |

**(b) Attack F1-score**

| Dataset | Attack | GAT | | | | GCN | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $G_0$ | $G_1$ | $G_2$ | DSV | $G_0$ | $G_1$ | $G_2$ | DSV |
| Facebook | Attack A | 0.806 | 0.778 | 0.702 | 0.070 | 0.830 | 0.768 | 0.718 | 0.076 |
| | Attack B | 0.816 | 0.790 | 0.713 | 0.069 | 0.857 | 0.815 | 0.723 | 0.089 |
| Pokec | Attack A | 0.733 | 0.690 | 0.622 | 0.074 | 0.730 | 0.706 | 0.606 | 0.083 |
| | Attack B | 0.791 | 0.703 | 0.672 | 0.079 | 0.807 | 0.744 | 0.674 | 0.088 |
| Spammer | Attack A | 0.774 | 0.602 | 0.589 | 0.123 | 0.716 | 0.638 | 0.545 | 0.114 |
| | Attack B | 0.768 | 0.605 | 0.591 | 0.118 | 0.685 | 0.596 | 0.540 | 0.097 |

and *average edge betweenness centrality* [49, 79]. The measurement of group size is trivial, and we omit the details. Besides these four properties, we considered the graph assortativity property [15, 53, 63], i.e., the tendency of connected nodes to have similar attributes. However, as we identified there only exists a weak correlation between assortativity and attack performance (more details are in Appendix B.3), we omit the details about the assortativity property.

**Group density (GD).** Essentially, the density of a given graph $\mathcal{G}$ is measured as the ratio of existing edges to all possible edges in $\mathcal{G}$ [77]. We adapt this to define the *group density* for each link group. In particular, given a link group $G \subseteq \mathcal{G}$, let $s_u$ and $s_v$ be the protected node features of $G$. Based on the relationship between $s_u$ and $s_v$, we consider two cases when calculating the density of $G$.

Case 1: $s_u = s_v$. For this case, we measure the group density as $Den(G) = \frac{2k_e}{k_n \times (k_n - 1)}$, where $k_e$ and $k_n$ respectively represent the number of edges and the number of nodes in $G$. Note that when $s_u = s_v$, the total number of possible edges in $G$ is $\frac{k_n(k_n-1)}{2}$. For example, given an M-M group that consists of three male nodes, the number of possible links between these nodes is 3.

Case 2: $s_u \neq s_v$. For this case, the density of $G$ is measured as $Den(G) = \frac{k_e}{k_i \times k_j}$, where $k_e$ is the number of edges in $G$, and $k_i$ and $k_j$ are the numbers of nodes associated with value $s_u$ and $s_v$ respectively. Note that when $s_u \neq s_v$, the total number of possible edges of $G$ is $k_i \times k_j$. For example, given an M-F group that consists of three male nodes and three female nodes, the number of possible links between these nodes is $3 \times 3 = 9$.

Conceptually, group density indicates how dense the group is in terms of edge connectivity. A group of high (low, resp.) GD indicates more (fewer, resp.) users in the group are connected. For example, in the Spammer dataset, the M-M group has the lowest density among the three subgroups. This indicates that both female and male users on Spammer social networks prefer to connect with female users than male ones.

**Average edge betweenness centrality (AEBC).** Centrality measures are important metrics for social network analysis to evaluate the structural importance of nodes and links. In this paper, we use *edge betweenness centrality* [28, 43], a widely-used centrality measurement, to measure the importance of edges. Edge betweenness centrality (EBC) measures the number of the shortest paths that go through an edge in a graph [27]. Typically, edges of high EBC are those "bridges" between nodes in the graph (i.e., removing them causes the graph to become disconnected). Given a link group $G$, the *average edge betweenness centrality* (AEBC) of $G$ is measured as the average EBC of all the edges in $G$. A link group of higher (lower, resp.) AEBC has more control over the graph as the graph's connectivity has higher (resp. lower) dependence on the group.

**Average node similarity (ANS).** Intuitively, node similarity measures the distance between any two nodes based on their neighborhood. In this paper, we consider *Jaccard similarity score* as the measurement of node similarity. Formally, given two nodes $u, v$, their Jaccard similarity $sim(u, v)$ is measured as: $sim(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|}$, where $N_u(N_v)$ is the neighborhood of $u$ ($v$). Intuitively, two nodes are more similar if they share more neighbors. Given a link group $G$, we measure its *average node similarity* (ANS) as the average of NS of all the connected node pairs $\{(u, v)\}$ in $G$. Intuitively, for a link group of high ANS, all edges in this group have similar neighborhoods.

## 5.2 Relationship between Structural Properties and Privacy Vulnerability

To gain a deeper understanding of the existence of DSV, we analyze the relationship between the attack performance and the four types of structural properties (group size, GD, ANS, and AEBC) through empirical analysis. In particular, given a training graph $\mathcal{G}$, we randomly sample 1,000 subgraphs from $\mathcal{G}$, where each subgraph contains $p\%$ nodes of $\mathcal{G}$. We tried four different settings of $p \in \{5\%, 10\%, 15\%, 20\%\}$. Then for each sampled subgraph, we generate a LMIA testing dataset. In particular, for each subgraph, we add its edges (members) into the testing data. Then we randomly sample the same number of disconnected node pairs (non-members) and add them to the testing dataset. Thus each LMIA testing dataset has equal number of members and non-members. For each sampled subgraph, we measure its structural properties of the subgraph as well as the attack performance (BAA and F1-score) of its corresponding testing data. From the results collected from 1,000 subgraph samples, we measure the Pearson correlation between each structural property and attack performance and analyze the causal effects of these structural properties on attack performance. In the following discussions, we first report the results of Pearson correlation (Section 5.2.1) followed by the results of causal effects (Section 5.2.2). Due to the limited space, we only present the results of Attack B. The results of Attack A can be found in Appendix B.3.

*5.2.1 Statistical Correlation Analysis.* Table 7 presents the Pearson correlation between the structural properties and attack performance. We have the following main findings from these results.

**Finding #3: No correlation between group size and attack performance.** Table 7 ("Group size" column) presents that the Pearson correlation between group size and attack performance is very weak. In particular, the correlation values fall in the range of [-0.04, 0.07] in all the settings. This is a surprising yet important finding, as, unlike the non-graph data where group size is one of the factors of vulnerability disparity [11, 44], group size does not impact attack performance for the graph data.

**Finding #4: A strong negative correlation exists between GD and attack performance.** Table 6 has indicated a correlation between group density and attack performance. We further measure their Pearson correlation. As shown in Table 7 ("GD" column), there exists a strong negative correlation between group density and attack performance. Specifically, *dense (sparse) groups are less (more) vulnerable to LMIA*. This is a key observation as it connects privacy vulnerability to a particular type of graph characteristics. We briefly explain why there exists a negative correlation between group density and attack performance by using Spammer dataset as an example. Recall that among the three subgroups (F-F, F-M, M-M) in the Spammer dataset, the M-M group has the lowest density. Intuitively, as the M-M group has low density, the male nodes are connected with fewer other nodes in the group at average than the F-M and F-F groups. As GNNs generate the node embeddings by aggregating from the neighborhood of these nodes, a node with a smaller (and more uniform) neighborhood will be associated with a more similar embedding to its neighbors than the one with a larger (and less uniform) neighborhood. Thus both embedding and posterior outputs of the male nodes in the sparse M-M group are more similar to their connected neighbors than the disconnected nodes in the same group. Therefore, the connected node pairs (members) and disconnected ones (non-members) in the M-M group are more distinguishable (either by their node similarity or posterior outputs, depending on which is used for the derivation of LMIA features) than those in the F-M and F-F groups. This explains why the links in the M-M group are more vulnerable to LMIA than those in the F-M and F-F groups (Finding #1). Detailed investigation of why there exists a correlation between group density and attack performance can be found in Appendix B.4.

**Finding #5: A strong negative correlation exists between AEBC and attack performance.** Table 7 ("AEBC" column) presents the Pearson correlation between AEBC and attack performance. We observe that a strongly negative correlation exists between AEBC and attack performance. In particular, the correlation always exceeds -0.5 in all settings and can be as high as -0.69. Due to the strong negative correlation, the subgroups with higher AEBC have lower attack performance. For example, we observe that in the Facebook dataset, the F-F group has the lowest AEBC value and the highest privacy vulnerability to LMIA. Intuitively, low AEBC values of the F-F edges indicate that few paths pass through these edges. Due to the message-passing mechanism of GNNs, the embeddings of the nodes on the F-F edges are aggregated from fewer neighbors than the nodes on the F-M and M-M edges. Therefore, the connected nodes are more distinguishable than the disconnected nodes in terms of node embeddings and posterior output in the F-F group than the other two groups. This makes the F-F group suffers from the highest privacy vulnerability. More detailed explanations of the reason behind this negative correlation can be found in Appendix B.5.

**Finding #6: A strong positive correlation exists between ANS and privacy leakage.** Table 7 ("ANS" column) presents a strong positive Pearson correlation between ANS and attack performance. The correlation is always higher than 0.54 and can be as high as 0.74. This strong positive correlation suggests that the groups of higher (lower, resp.) average node similarity are more (less, resp.) vulnerable to the attack. This correlation can be explained from the nature of GNNs and the principle of LMIA attacks: the nodes that have more similar neighbors will have more similar embeddings and thus more similar posterior outputs. As LMIA distinguishes members and non-members based on the similarity of node embeddings and posterior outputs, it should be more accurate to attack the groups that have higher ANS. A more detailed explanation of this correlation can be found in Appendix B.6.

*5.2.2 Causal Analysis.* So far we have shown the existence of a strong Pearson correlation between density/AEBC/ANS and privacy vulnerability. However, Pearson correlation does not imply that the difference in the density/AEBC/ANS of different subgroups *causes* DSV. To further examine the root cause of DSV, we further investigate whether there exists a causal relationship between the three graph properties of the subgroups and the privacy vulnerability of these subgroups.

**Counterfactual causality.** Causal analysis is a well-known method for root cause analysis. Essentially, a standard dataset for causal analysis includes the feature matrix $X$, a vector of treatments

**Table 7: Pearson correlation between attack performance and structural properties (Attack B). GD: group density; AEBC: average group betweenness centrality; ANS: average group node similarity.**

| Dataset | GNN | Balanced Attack Accuracy (BAA) | | | | Attack F1-score | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Group size | GD | AEBC | ANS | Group size | GD | AEBC | ANS |
| Facebook | GAT | -0.04 | -0.68 | -0.61 | 0.66 | 0.05 | -0.74 | -0.69 | 0.68 |
| | GCN | 0.03 | -0.63 | -0.58 | 0.61 | 0.07 | -0.67 | -0.53 | 0.64 |
| Pokec | GAT | -0.01 | -0.74 | -0.63 | 0.65 | 0.08 | -0.71 | -0.67 | 0.62 |
| | GCN | 0.03 | -0.64 | -0.64 | 0.61 | 0.06 | -0.65 | -0.67 | 0.65 |
| Spammer | GAT | 0.04 | -0.74 | -0.51 | 0.54 | 0.07 | -0.70 | -0.59 | 0.66 |
| | GCN | -0.02 | -0.63 | -0.54 | 0.60 | 0.03 | -0.57 | -0.61 | 0.62 |

**Table 8: Counterfactual causal effect of structural properties on attack performance of Attack B (GD: group density; AEBC: average group betweenness centrality; ANS: average group node similarity).**

| Dataset | GNN | Balanced Attack Accuracy (BAA) | | | Attack F1-score | | |
|---|---|---|---|---|---|---|---|
| | | GD | AEBC | ANS | GD | AEBC | ANS |
| Facebook | GAT | -0.45 | -0.12 | 0.25 | -0.48 | -0.18 | 0.31 |
| | GCN | -0.43 | -0.11 | 0.32 | -0.45 | -0.15 | 0.25 |
| Pokec | GAT | -0.39 | -0.13 | 0.27 | -0.36 | -0.13 | 0.28 |
| | GCN | -0.32 | -0.11 | 0.31 | -0.35 | -0.14 | 0.32 |
| Spammer | GAT | -0.43 | -0.12 | 0.34 | -0.46 | -0.14 | 0.20 |
| | GCN | -0.39 | -0.19 | 0.27 | -0.44 | -0.18 | 0.22 |

$t$, and outcome $y$. To understand the causal effect of the treatment $t$ on the outcome $y$, one solution is to use the *counterfactual outcome* to quantify the change of the outcome $y$ when the value of the treatment $t$ is manipulated (e.g., change from $t = 1$ to $t = 0$). The magnitude of the causal effect can be measured by *Average Treatment Effect* (ATE) [57]:

$$ATE = \mathbb{E}[y = 1|t = 1] - \mathbb{E}[y = 1|t = 0], \tag{8}$$

where $\mathbb{E}[\cdot]$ denotes the expectation. ATE falls in the range $[-1, 1]$, where a positive (negative, resp.) ATE value indicates that the treatment $t = 1$ ($t = 0$, resp.) is the cause of the outcome $y = 1$. Furthermore, a larger ATE value indicates higher causality of treatment $t$ on the outcome $y$.

**Measuring counterfactual causality of structural properties.** As ATE only considers binary treatment and outcome values, but both structural property values and attack performance values are in a continuous domain, we convert both structural property value and attack performance into the binary domain. Specifically, we say a link subgroup $G$ in the *treatment group* (denoted as $t = 1$) if its density (ANS/AEBC, resp.) is larger than that of the whole graph. Otherwise, $G$ is considered as in the *control group* (denoted as $t = 0$). Similarly, $G$'s outcome is denoted as $y = 1$ if its attack performance is better than that of the whole graph, and $y = 0$ otherwise.

To measure the causal effect of a specific structural property (GD/ANS/AEBC) on attack performance, we first sample 200 subgraphs from the training graph. For each sampled subgraph, we measure the structural property (GD/ANS/AEBC) and attack performance (BAA and F1-score) for all of its link groups. Then for each link group $G$, we determine if it belongs to a control group or a treatment group (i.e., $t = 0/1$) based on its structural property. Finally, we generate the *counterfactual* $G'$ of $G$ by adding/removing edges to/from $G$, so that the treatment value $t$ of $G'$ is opposite to that of $G$. We measure the attack performance of $G'$, and label the outcome $y$ of both $G$ and $G'$. After we have collected the outcome values of all link subgroups and their counterfactuals, we measure ATE (Eqn. (8)).

**Finding #7: GD/AEBC/ANS has causal effect on privacy vulnerability.** Table 8 presents the ATE of graph density ("GD" column) on attack performance. We observe negative ATE values in the range [-0.32, -0.48]. This indicates that density has a causal effect on attack performance, where the low density (i.e., $t = 0$) of the subgroups is the cause of high LMIA performance (i.e., $y = 1$). This is also consistent with the observed negative Person correlation between group density and attack performance (Finding #4).

Table 8 also presents ATE of both AEBC and ANS. We observe negative ATE values for AEBC and positive ATE values for ANS, which fall in the range of [-0.11, -0.19] and [0.20, 0.32] respectively. This suggests that both AEBC and ANS have causal effects on attack performance. The signs of ATE values show that the subgroups of low AEBC (i.e., t=0) and high ANS (i.e., t=1) are the causes of high attack performance (i.e., y=1). These observations are also consistent with the observed Person correlations between AEBC/ANS and attack performance (Findings # 5 and #6).

**Finding #8: The difference between GD of subgroups has the largest causal effect on DSV.** As shown in Table 8, density presents the highest ATE values among the three properties, while AEBC witnesses the lowest ones. This is also consistent with the Pearson correlation results where density witnesses the strongest Pearson correlation and AEBC witnesses the weakest Pearson correlation among the three properties. Hence, we believe that the

difference in the density of different subgroups plays a more important role in DSV than AEBC and ANS.

# 6 MITIGATING DISPARITY IN SUBGROUP VULNERABILITY

In this section, we aim to answer research question *RQ3 — How to mitigate the disparity in subgroup vulnerability and provide fair privacy protection against LMIA across all subgroups*? According to the findings presented in Section 5, DSV is correlated to the disparity between subgroups on the properties of GD, AEBC, and ANS. Thus, we will be able to mitigate DSV by reducing those disparities. Considering that GD has the most significant causal effect on DSV (**Finding #8**), we focus on designing mitigation methods to reduce the gap in GD.

A naive method of reducing the gap between GD is to add/remove edges for different groups in the training graph in one shot so that all subgroups have the same density. However, this method can require adding/removing a substantial amount of edges when handling graphs that have a skewed distribution of GD, thus incurring high accuracy loss of the target model. To this end, we design a defense method named FairDefense to reduce GD disparity without heavily compromising the target model accuracy. The key idea is to "flatten" the density of all subgroups along the training of the target model, instead of doing so in one shot. In particular, FairDefense perturbs only a small portion of edges in the training graph at each iteration while training the target model, so that the gap between the density of all subgroups is reduced in multiple rounds. Eventually, the density of all subgroups becomes close to each other at the end of training. By distributing the perturbation over multiple iterations, the accuracy loss of the target model is largely alleviated.

## 6.1 Design of FairDefense

At a high level, FairDefense provides defense power against LMIA attack by introducing perturbation on the training graph through randomization: *each node pair is added to the adjacency matrix A with some probability before A is passed to the MSG() function (Eqn. (1)) at each iteration of the training process.* As both GAT and GCN models use the same $MSG()$ function, FairDefense can be easily integrated into both of them.

Specifically, at each iteration of the training process of the target model, each node pair $(u, v)$ where $e(u, v) \in \mathcal{G}$ (i.e., a member) will be randomized as either a member (i.e., $A[u, v] = 1$) or a non-member (i.e., $A[u, v] = 0$) with a given probability:

$$A[u,v] = \begin{cases} 1, & \text{with prob. } 1 - p_1; \\ 0. & \text{with prob. } p_1. \end{cases} \tag{9}$$

where

$$p_1 = \frac{k_o}{e^{\gamma} k_m + k_o}, \tag{10}$$

with $k_m$ and $k_o$ being the number of member and non-member edges in $\mathcal{G}$ respectively, and $\gamma > 0$ the privacy parameter for defense against the attack.

Similarly, for each node pair $(u, v)$ where $e(u, v) \notin \mathcal{G}$ (i.e., a non-member), it will be randomized as either a member or a non-member with the following probability:

$$A[u,v] = \begin{cases} 0, & \text{with prob. } 1 - p_2; \\ 1. & \text{with prob. } p_2. \end{cases} \tag{11}$$

where

$$p_2 = \frac{k_m}{e^{\gamma} k_m + k_o}. \tag{12}$$

Intuitively, higher $\gamma$ will lead to lower $p_1$ and $p_2$, and thus lower perturbation, which consequently leads to less defense against LMIA. However, higher $\gamma$ will also lead to less DSV mitigation. We will discuss the trade-off between DSV mitigation defense later in this section.

As FairDefense always adds perturbation to the original graph instead of the graph that has been noised in the previous iterations, the perturbation in the previous iterations is abandoned. Hence, the impact of the perturbation on the target model will not be accumulated through the iterations. Our empirical study will show that, by doing so, FairDefense can achieve better target model accuracy than continuous perturbation on the training graph over the iterations.

**DSV mitigation by FairDefense.** Recall that one of the underlying reasons for DSV is the disparity in the density of subgroups (Finding #8). Can FairDefense reduce the gap between the density of all subgroups and thus mitigate DSV? To answer this question, we present the following theorem to show that FairDefense guarantees to mitigate the gap between the density of different subgroups.

Theorem 1. *Given a graph $\mathcal{G}$ and any two subgroups $G_i, G_j \in \mathcal{G}$, let $d_i$ and $d_j$ be the original density of $G_i$ and $G_j$. Also let $E(d_i)$ and $E(d_j)$ be the expected density of $G_i$ and $G_j$ by FairDefense. Then the gap between the density of $G_i$ and $G_j$ (denoted as $|E(d_i) - E(d_j)|$) is as follows:*

$$|E(d_i) - E(d_j)| = |d_i - d_j|(1 - \frac{k_m + k_o}{e^{\gamma} k_m + k_o}) \tag{13}$$

The proof of Theorem 1 can be found in Appendix C.1. With any $\gamma > 0$, it should hold that $0 < \frac{k_m + k_o}{e^{\gamma} k_m + k_o} < 1$. Thus $|E(d_i) - E(d_j)| < |d_i - d_j|$. In other words, the gap between the density of any two groups $G_i$ and $G_j$ is reduced after the deployment of FairDefense.

Following Eqn. (13), larger $\gamma$ will lead to a smaller gap between the density of different groups (i.e., $|E(d_i) - E(d_j)|$). Ideally, FairDefense achieves the best DSV mitigation effect when all link groups have the same expected density (i.e., $|E(d_i) = E(d_j)|$). However, according to Eqn. (13), $|E(d_i) = E(d_j)|$ holds only when $\gamma = 0$, which will lead to the maximum amounts of perturbation on the graph. Naturally, this leads to the trade-off between DSV mitigation and target model accuracy. We will show the performance of FairDefense in the trade-off between DSV mitigation and target model accuracy through empirical evaluation (Section 6.2).

**Preservation of graph characteristics by FairDefense.** As the perturbation of FairDefense changes the density of each link group, does it destroy the graph characteristics, for example, the density of the whole graph? To answer this question, we study the impact of FairDefense on graph density.

First, we have the following lemma to show the probability that a node pair will be a member/non-member edge after perturbation.

Lemma 2. *Consider a graph $\mathcal{G}$ and any node pair $(u, v) \in \mathcal{G}$, the probability $p_m$ that $(u, v)$ is connected (i.e., a member edge) at any iteration is*

$$p_m = \frac{k_m}{k_m + k_o}, \tag{14}$$

*And the probability $p_n$ that $(u, v)$ is not connected (i.e., a non-member edge) at any iteration is*

$$p_n = \frac{k_o}{k_m + k_o}. \tag{15}$$

The proof of Lemma 2 can be found in Appendix C.2. As suggested by Lemma 2, a node pair is less likely to be a member edge than a non-member (i.e., $p_m < p_n$) in sparse graphs (i.e., $k_m < k_o$) after perturbation, and the opposite in the dense graphs. Based on Lemma 2, we have the following proof to show that FairDefense preserves the density of the training graph.

Theorem 3. *Consider a graph $\mathcal{G}$ and any node pair $(u, v) \in \mathcal{G}$, it must hold that $E(d) = d$, where $E(d)$ and $d$ are the expected density of $\mathcal{G}$ by FairDefense and the original density of $\mathcal{G}$ respectively.*

The proof of Theorem 3 can be found in Appendix C.3. As graph density is a key type of graph characteristic, preservation of the graph density will enable various types of density-based graph analytics [46, 62].

## 6.2 Performance of FairDefense

*6.2.1 Experimental Setup.* We first explain the setup of empirical evaluation for this part of experiments.

**Evaluation metrics.** We use the same target model accuracy metric as described in Section 3.2.1: the accuracy of node classification. Ideally, the defense should harm the target model performance as little as possible while providing both protection against LMIA and DSV mitigation. In the evaluation of defense effectiveness, we measure both balanced attack accuracy (*BAA*) and F1-score (F1) after the deployment of the defense. We further measure DSV after the deployment of the defense mechanisms.

**Baseline methods.** We consider four different baseline methods and categorize these four methods into two types based on their working mechanisms:

- **Differential privacy (DP) based baselines.** *Differential privacy* (DP) [23] has been considered as a de facto standard for data privacy. We consider two types of DP that are deployed at different granularity as the baseline: (i) Global DP (**DP-SGD**): We consider the well-known DP-SGD method [1] that adds Laplace noise to the stochastic gradient descent (SGD) during the training process of the target model to provide global DP; (ii) Local DP (**RR**): We consider the algorithm presented in [40] to provide local DP. It randomizes the neighbor lists of all nodes by flipping the absence/presence bit (0/1) of each node in the list with the probability $\frac{1}{1+e^{\epsilon}}$, where $\epsilon$ denotes the privacy budget.
- **Perturbation-based baselines.** We consider two alternative ways to add perturbation on the training graph: (1) **One-shot perturbation (OP)**: instead of perturbation at each iteration, the training graph is perturbed only once by OP before the training process starts; (2) **Accumulated perturbation (AP)**: instead of abandoning the perturbation added by the previous iterations, AP perturbs the training graph that has been randomized in the previous iterations. Thus, the perturbation is accumulated during the iterations.

As FairDefense does not provide any theoretical privacy/utility guarantee, making theoretical-level comparisons with the baselines (e.g., the DP-based one) is infeasible. Therefore, we provided empirical comparisons between FairDefense and the baselines instead.

*6.2.2 Evaluation Results.* To ensure a fair comparison between FairDefense and the four baselines, we empirically configure the privacy parameters of FairDefense and the baselines in the way that their corresponding target models have similar accuracy (in the range [0.552, 0.563] for GAT and [0.569, 0.581] for GCN).

**Defense effectiveness.** Table 9 ("Attack Acc." column) reports the attack accuracy (BAA as the metric) of Attack A before and after defense for both Spammer and Facebook graphs. We can see that FairDefense is effective in defending LMIA by reducing the attack accuracy significantly. Take Spammer dataset for example, the accuracy drops from 0.687 to 0.490 for the GAT model, and from 0.641 to 0.514 for the GCN model. For the Facebook dataset, the attack accuracy against two models drops from 0.734 and 0.810 to 0.651 and 0.621 respectively. Furthermore, FairDefense outperforms the four baselines in terms of the defense power for all the settings. Consider the GCN model as an example. The four baseline methods only can reduce the attack accuracy to around 0.558 on the Spammer dataset and 0.729 on the Facebook dataset respectively. In contrast, FairDefense can reduce the attack accuracy to 0.514 on the Spammer dataset and 0.621 on the Facebook dataset respectively, which is significantly better than the four baselines. For example, on the Facebook dataset, FairDefense is 4%-26% more effective in defense and 14%-55% more effective in DSV mitigation than the baselines while maintaining high target model accuracy. We also observe that, unexpectedly, DP-SGD fails to provide protection against LMIA. We explain the reason in Appendix D.1.

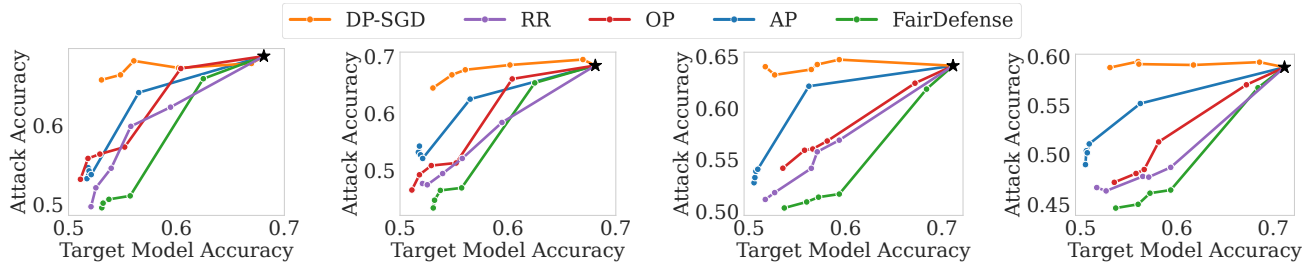
One interesting observation from Table 9 is that the attack accuracy is reduced to around 0.5 for most of the cases. This raises the question that whether DSV is mitigated because the attack became ineffective. Our answer is negative — the mitigation effect is not necessarily caused by attack ineffectiveness. Consider Figure 1 (b) and Figure 2 (b). When the target model accuracy stays around 65%, the attack accuracy remains high (Figure Figure 1 (b)), but DSV is mitigated by about 35% (Figure 2 (b)).

**DSV mitigation.** Table 9 ("DSV" column) reports the amounts of DSV before and after the deployment of FairDefense and the four baselines. Overall, FairDefense mitigates DSV effectively. For example, FairDefense reduces DSV from 0.14 to 0.038 for GAT model, and from 0.15 to 0.044 for GCN model under Attack A. In addition, FairDefense largely outperforms the four baseline methods in DSV mitigation. For example, consider Attack A on GAT. The baselines only reduce DSV to no smaller than 0.063 while FairDefense can reduce DSV to 0.038.

Due to limited space we include the result of attack accuracy and DSV of Attack B in Appendix D.2. Similar to the observation on Attack A, FairDefense still outperforms the baselines on both defense effectiveness and DSV mitigation.

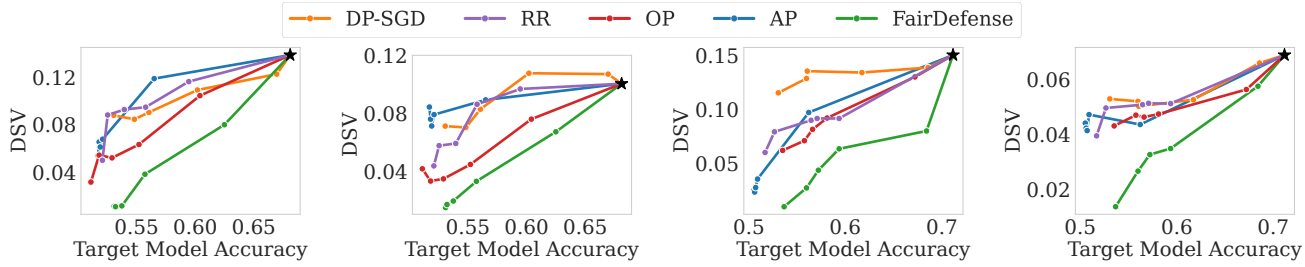
**Table 9: DSV mitigation and defense effectiveness of FairDefense and four baselines (Spammer and Facebook graphs, Gender as the protected node feature, BAA as the attack accuracy metric, Attack A as the attack type). The lowest BAA and the lowest DSV are marked with pink and green respectively. For a fair comparison, we choose the privacy parameters of FairDefense and the baselines that have similar target model accuracy ("Target Acc." column).**

| Dataset | Defense | GAT | | | GCN | | |
|---|---|---|---|---|---|---|---|
| | | Target Acc. | Attack Acc. | DSV | Target Acc. | Attack Acc. | DSV |
| Spammer | No defense | 0.681 | 0.687 | 0.139 | 0.712 | 0.641 | 0.150 |
| | DP-SGD | 0.561 | 0.681 | 0.091 | 0.569 | 0.622 | 0.138 |
| | RR | 0.558 | 0.598 | 0.105 | 0.571 | 0.558 | 0.129 |
| | OP | 0.552 | 0.572 | 0.063 | 0.581 | 0.568 | 0.092 |
| | AP | 0.565 | 0.637 | 0.109 | 0.562 | 0.621 | 0.097 |
| | FairDefense | 0.563 | 0.490 | 0.038 | 0.572 | 0.514 | 0.044 |
| Facebook | Original | 0.812 | 0.734 | 0.083 | 0.857 | 0.810 | 0.093 |
| | DP-SGD | 0.721 | 0.753 | 0.074 | 0.678 | 0.785 | 0.087 |
| | RR | 0.715 | 0.677 | 0.079 | 0.667 | 0.729 | 0.089 |
| | OP | 0.714 | 0.720 | 0.073 | 0.677 | 0.762 | 0.075 |
| | AP | 0.726 | 0.714 | 0.059 | 0.668 | 0.755 | 0.081 |
| | FairDefense | 0.733 | 0.651 | 0.051 | 0.673 | 0.621 | 0.040 |




(a) Attack A on GAT (b) Attack B on GAT (c) Attack A on GCN (d) Attack B on GCN

**Figure 1: Trade-off between defense and target model accuracy (Spammer dataset, BAA as the attack accuracy metric). The ⋆ mark denotes no defense. The defense with lower attack accuracy and higher target model accuracy has a better trade-off.**




(a) Attack A on GAT (b) Attack B on GAT (c) Attack A on GCN (d) Attack B on GCN

**Figure 2: Trade-off between DSV and target model performance (Spammer dataset). The defense with lower DSV and higher target model accuracy has a better trade-off.**

**Trade-off between defense and target model accuracy.** Privacy comes at the cost of reducing target model accuracy. To visualize the trade-off between defense effectiveness and target accuracy loss, we generate the *attack-utility curve* in which each point presents a pair of attack accuracy and target model accuracy (utility) values. To generate the curve, we vary the privacy parameters of FairDefense and the four baselines, and measure both attack accuracy (BAA as the metric) and target model accuracy for each privacy parameter value. Intuitively, a method that has lower attack accuracy and higher target model accuracy has a better trade-off between defense and target model accuracy. Due to limited space, we only present the results of the Spammer graph as it has the highest DSV (as shown in Table 6). The results for the other two datasets can be found in Appendix D.3.

Figure 1 presents the attack-utility curve generated from the Spammer dataset. Intuitively, the defense method that has higher defense strength (i.e. lower attack accuracy) and aligned target model accuracy with the others has a better trade-off between defense and target model accuracy. We observe that FairDefense outperforms the four baselines in the trade-off between defense effectiveness and target model accuracy: FairDefense achieves lower attack accuracy than the baselines under the same target model accuracy, while it delivers higher target model accuracy than the baselines when they have the same attack accuracy. The result of the Facebook dataset is similar and can be found in Appendix D.3.

**Trade-off between DSV mitigation and target model accuracy.** Besides the trade-off between defense and target model accuracy, there also exists a trade-off between defense and DSV mitigation (Theorem 1). To visualize the trade-off between DSV mitigation and target model accuracy, we generate a *DSV-utility curve* in which each point presents a pair of DSV and target model accuracy values. We pick the same parameters used in the measurement of the trade-off between defense and target model accuracy (Figure 1).

Figure 2 presents the DSV-utility curve generated from the Spammer dataset. We vertically visualize the comparison of the DSV of different methods when they present aligned target model accuracy. Intuitively, the defense method that has higher DSV and aligned target model accuracy with the others has a better trade-off between DSV and target model accuracy. We observe that FairDefense better addresses the trade-off between DSV mitigation and target model accuracy than the four baselines: FairDefense has lower DSV than the baselines when they have the same target model performance, and it achieves higher target model performance than the four baselines when they deliver the same DSV.

## 7 RELATED WORK

**Membership inference attacks.** Membership inference attack (MIA) against the ML models was initially introduced by [64]. Recent works [48, 51, 59, 67, 73] investigated the factors that affect the performance of MIA. [48, 73] demonstrated that overfitting contributes to information leakage but is not the fundamental factor. [67] claimed that MIA is data-driven but dominated by the target model. [59, 67] show that the attack models are largely transferable. New MIA models have been developed to attack Federated Learning [52], collective learning [51], generative adversarial networks (GANs) [12, 33], embedding models [65] and GNNs [34, 35, 37]. We refer the audience to some recent surveys [5, 37] on MIA and its related studies. In this paper, we consider LMIA against GNNs.

**Privacy inference attacks against GNNs.** Privacy leakage of GNNs has been investigated from several aspects. The existing attacks can be categorized into the following types: (1) *Membership inference attacks* [21, 34, 35, 54, 72] that infer the existence of certain nodes [35], edges [21, 34, 72], and subgraphs [71] in the training graph; (2) *Property inference attacks* [70, 76] that infer the specific properties of graphs, such as density, number of nodes and the distribution of nodes in the training graph; (3) *Model extraction attacks* [61, 75] that reconstruct a GNN model which has behaviors to the given target model; and (4) *Attribute inference attacks* [21] that infer the sensitive attributes in the training graph. In this paper, we consider link-level membership inference attacks [34], and focus on the disparate vulnerability of subgroups against this attack.

**Algorithmic fairness and privacy.** Recent research [31, 41, 58, 66] explored how to achieve fairness and privacy independently. Meanwhile, a parallel line of research [22, 24] considered the interaction between fairness and privacy in ML. Dwork et al. [22] indicated that differential privacy (DP) techniques can be adapted to satisfy fairness in ML. Several recent works have investigated the disparity in the attacks and privacy of ML models. Bagdasaryan et al. [3] display the disparate impact of DP. Their empirical evaluation shows that differentially private models have a larger accuracy reduction on the underrepresented groups. Dibbo et al. [17] investigated disparate vulnerability in model inversion attacks. In terms of vulnerability disparity of membership inference attacks, Kulynych et al. [44] identified the existence of disparate vulnerability across different groups against MIA over tabular data. Da et al. [78] investigated the disparity in both privacy vulnerability of membership inference attacks and protection power of the existing MIA defense mechanisms across different groups. While both works [44, 78] considered the conventional classification models over non-graph data as the target model, we consider GNNs. Our work is novel as: (1) we are the first to investigate the disparate effects of LMIA against GNNs; (2) we identify three unique graph properties as the underlying reasons for DSV of GNN; and (3) we design the first LMIA defense that can mitigate DSV.

## 8 CONCLUSION

In this paper, we investigate the disparate effects of LMIAs on different link subgroups when attacking GNNs. First, we perform extensive empirical evaluation over three real-world social network graphs and two representative GNN models. Our empirical results demonstrate the existence of non-negligible DSV in all the examined settings. Second, we identify the causal effects of three types of graph structural properties on DSV. Third, based on the analysis, we design a new defense mechanism named FairDefense that not only defends against the LMIA attack but also mitigates DSV across subgroups by randomizing the training graph during the training of the target model. Our experimental results demonstrate the effectiveness of FairDefense in terms of defense, DSV mitigation, and the trade-off between defense and target model accuracy.

**Future work.** First, beyond the four graph structural properties that we have investigated in the paper, other properties such as graph homophily [50] and other centrality measurements such as degree centrality and closeness centrality [47, 69] may impact DSV. We will investigate the correlations between these properties and DSV in future work. Another interesting direction is to understand the theoretical privacy guarantee (e.g., local differential privacy [16]) of FairDefense. Another research direction is to investigate DSV for other types of inference attacks against GNNs (e.g., node-level membership inference attacks [21, 35]).

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This project was supported by the National Science Foundation (#CNS-2029038; #CNS-2135988). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

# REFERENCES


[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of SIGSAC Conference on Computer and Communications Security*, 2016.

[2] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. walk2friends: Inferring social links from mobility profiles. In *Proceedings of SIGSAC Conference on Computer and Communications Security*, 2017.

[3] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, 2019.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*, 2015.

[5] Yang Bai, Ting Chen, and Mingyu Fan. A survey on membership inference attacks against machine learning. *International Journal of Network Security*, 2021.

[6] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California Law Review*, 2016.

[7] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of Conference on Fairness, Accountability and Transparency*, 2018.

[8] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 2010.

[9] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. Membership inference attacks from first principles. *IEEE Symposium on Security and Privacy*, 2021.

[10] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[11] Hongyan Chang and Reza Shokri. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 292–303, 2021.

[12] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of Conference on Computer and Communications Security*, 2020.

[13] Jianfeng Chi, Yuan Tian, Geoffrey J Gordon, and Han Zhao. Understanding and mitigating accuracy disparity in regression. In *Proceedings of International Conference on Machine Learning*, 2021.

[14] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, 2021.

[15] Manvi Choudhary, Charlotte Laclau, and Christine Largeron. A survey on fairness for machine learning on graphs. *arXiv preprint arXiv:2205.05396*, 2022.

[16] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[17] Sayanton V. Dibbo, Dae Lim Chung, and Shagufta Mehnaz. Model inversion attack with least information and an in-depth analysis of its disparate vulnerability. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2023.

[18] Yushun Dong, Jian Kang, Hanghang Tong, and Jundong Li. Individual fairness for graph neural networks: A ranking based approach. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 300–310, 2021.

[19] Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM Web Conference 2022*, pages 1259–1269, 2022.

[20] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[21] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying Privacy Leakage in Graph Embedding. In *EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2020.

[22] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of Innovations in Theoretical Computer Science conference*, 2012.

[23] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of Theory of Cryptography Conference*, 2006.

[24] Michael D Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. Privacy for all: Ensuring fair and equitable privacy protections. In *Proceedings of Conference on Fairness, Accountability and Transparency*, 2018.

[25] Shobeir Fakhraei, James Foulds, Madhusudana Shashanka, and Lise Getoor. Collective spammer detection in evolving multi-relational social networks. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2015.

[26] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of SIGSAC conference on Computer and Communications Security*, 2018.

[27] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 2002.

[28] K-I Goh, Eulsik Oh, Byungnam Kahng, and Doochul Kim. Betweenness centrality correlation in social networks. *Physical Review E*, 2003.

[29] Neil Zhenqiang Gong and Bin Liu. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *Proceedings of USENIX Security Symposium*, 2016.

[30] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings of International Joint Conference on Neural Networks*, 2005.

[31] Sara Hajian, Josep Domingo-Ferrer, Anna Monreale, Dino Pedreschi, and Fosca Giannotti. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 2015.

[32] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 2016.

[33] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies*, 2019.

[34] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *Proceedings in USENIX Security Symposium*, 2021.

[35] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429*, 2021.

[36] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 519–535, 2020.

[37] Hongsheng Hu, Zoran Salcic, Sun Lichao, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys*, 2021.

[38] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. *arXiv preprint arXiv:2101.01341*, 2021.

[39] Hussain Hussain, Meng Cao, Sandipan Sikdar, Denis Helic, Elisabeth Lex, Markus Strohmaier, and Roman Kern. Adversarial inter-group link injection degrades the fairness of graph neural networks. *arXiv preprint arXiv:2209.05957*, 2022.

[40] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. Locally differentially private analysis of graph statistics. In *Proceedings of USENIX Security Symposium*, 2021.

[41] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. In *Proceedings of International Conference on Machine Learning*, 2019.

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

[43] Nicolas Kourtellis, Tharaka Alahakoon, Ramanuja Simha, Adriana Iamnitchi, and Rahul Tripathi. Identifying high betweenness centrality nodes in large social networks. *Social Network Analysis and Mining*, 2013.

[44] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. Disparate vulnerability to membership inference attacks. In *Privacy-Enhancing Technologies Symposium*, 2021.

[45] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

[46] Hao Li, Xiaojie Liu, Tao Li, and Rundong Gan. A novel density-based clustering algorithm using nearest neighbor graph. *Pattern Recognition*, 2020.

[47] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. On generalized degree fairness in graph neural networks. *arXiv preprint arXiv:2302.03881*, 2023.

[48] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[49] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:1048–1061, 2021.

[50] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

[51] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *Proceedings of Symposium on Security and Privacy*, 2019.

[52] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of Symposium on Security and Privacy*, 2019.

[53] M. E. J. Newman. Mixing patterns in networks. 2003.

[54] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *Proceedings of International Conference on*

*Trust, Privacy and Security in Intelligent Systems, and Applications*, 2021.
[55] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, Linlin Chen, Taeho Jung, and Junze Han. Social network de-anonymization and privacy inference with knowledge graph model. *IEEE Transactions on Dependable and Secure Computing*, 2019.
[56] Shahbaz Rezaei and Xin Liu. On the difficulty of membership inference attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7900, 2021.
[57] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 1974.
[58] Salvatore Ruggieri, Sara Hajian, Faisal Kamiran, and Xiangliang Zhang. Anti-discrimination analysis using privacy attack strategies. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2014.
[59] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of The Network and Distributed System Security Symposium*, 2018.
[60] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 2009.
[61] Yun Shen, Xinlei He, Yufei Han, and Yang Zhang. Model stealing attacks against inductive graph neural networks. In *Proceedings of Symposium on Security and Privacy*, 2021.
[62] Hiroaki Shiokawa, Tomokatsu Takahashi, and Hiroyuki Kitagawa. Scalescan: scalable density-based graph clustering. In *Database and Expert Systems Applications International Conference*, 2018.
[63] Yasuhiro Shirata. The evolution of fairness under an assortative matching rule in the ultimatum game. *International Journal of Game Theory*, 2012.
[64] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of Symposium on Security and Privacy*, 2017.
[65] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of Conference on Computer and Communications Security*, 2020.
[66] Cuong Tran, Ferdinando Fioretto, and Pascal Van Hentenryck. Differentially private and fair deep learning: A lagrangian dual approach. *Proceedings of Association for the Advancement of Artificial Intelligence Conference*, 2020.
[67] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.
[68] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of International Conference on Learning Representations*, 2018.
[69] Xiaofeng Wang, Xiaojie Chen, and Long Wang. Evolutionary dynamics of fairness on graphs with migration. *Journal of Theoretical Biology*, 2015.
[70] Xiuling Wang and Wendy Hui Wang. Group property inference attacks against graph neural networks. 2022.
[71] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Adapting membership inference attacks to gnn for graph classification: Approaches and implications. In *Proceedings of International Conference on Data Mining*, 2021.
[72] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *Proceedings of Symposium on Security and Privacy*, 2022.
[73] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of Computer Security Foundations Symposium*, 2018.
[74] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of International Conference on World Wide Web*, 2017.
[75] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. Graphmi: Extracting private graph data from graph neural networks. In *Proceedings of International Joint Conferences on Artificial Intelligence*, 2021.
[76] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. In *Proceedings of USENIX Security Symposium*, 2022.
[77] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *Privacy, Security, and Trust in KDD*, 2008.
[78] Da Zhong, Haipei Sun, Jun Xu, Neil Gong, and Wendy Hui Wang. Understanding disparate effects of membership inference attacks and their countermeasures. In *Proceedings of Asia Conference on Computer and Communications Security*, 2022.
[79] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 2008.

**Table 10: Setup of the target models.**

| Setting | | Facebook | Pokec | Spammer |
|---|---|---|---|---|
| GAT | # of heads | 8 | 6 | 4 |
| | # of nodes per layer | 8 | 6 | 4 |
| | # of labels | 4 | 2 | 2 |
| | Weight decay | 0.1 | 0.1 | 0.1 |
| | Learning rate | 0.004 | 0.001 | 0.005 |
| GCN | # of nodes per layer | 32 | 32 | 20 |
| | # of labels | 4 | 2 | 2 |
| | Weight decay | 0.001 | 0.005 | 0.005 |
| | Learning rate | 0.001 | 0.005 | 0.005 |

# APPENDIX

# A ADDITIONAL DETAILS OF EXPERIMENTAL SETUP

## A.1 Details of the Datasets

We use the following three social network graph datasets in the experiments: (1) **Facebook** graph[4] represents the social relationship of Facebook users. Each node in graph denotes a user and each edge denotes a friend relationship between two nodes (users). The node features include age, gender, education, etc. (2) **Pokec** social network graph[5] consists of users in Pokec, the most popular online social network in Slovakia. Each node in this graph denotes a Pokec user and each edge denotes the friend relationship between two nodes (users). Each node has user demographic features such as age, gender, hobbies, etc. (3) **Spammer** dataset is a social graph collected from *Tagged.com*[6], an online social network platform. Each node in this graph denotes a user and each edge represents a friend relationship between two nodes (users). The original dataset [25] contains 5.6 million users and 858 million links between the users. The task is to identify the *spammers* who are the users with malicious behaviors like sharing spam messages and sending fraudulent links. We sample 10,000 nodes from the original graph with the edges restored.

## A.2 Parameter Setup of Target Models

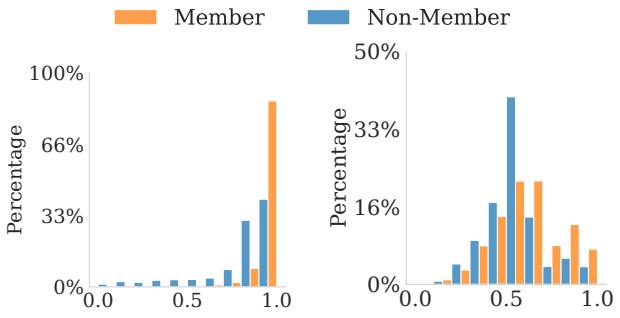
Table 10 shows the parameter setup of GCN and GAT models.

# B ADDITIONAL RESULTS OF ATTACK PERFORMANCE

## B.1 Explanation of Why Attack A Has Better Performance than Attack B on Spammer Dataset

Intuitively, as Attack A only considers the posterior outputs that encode structural similarity, while Attack B considers both structural similarity and node-feature similarity, Attack A will have better performance than Attack B only when the distribution of posterior similarity between members and non-members is inconsistent with

[4]Facebook dataset: https://snap.stanford.edu/data/ego-Facebook.html.
[5]Pokec dataset: https://snap.stanford.edu/data/soc-pokec.html
[6]Tagged dataset:https://linqs-data.soe.ucsc.edu/public/social_Tagged/




(a) Node Posterior Similarity (b) Node Feature Similarity

**Figure 3: Distribution of node feature similarity and node posterior similarity of members and non-members (GCN as target model, Spammer dataset)**

that of node-feature similarity. Following this reasoning, we measure both node-feature similarity and posterior similarity between the node pairs, where the normalized L2 distance is used as the similarity function.

Figure 3 illustrates the distribution of both posterior similarity and node-feature similarity on Spammer dataset. We observe that, while members and non-members are distinguishable by their posterior similarity (Figure 3 (a)), they become indistinguishable by their node-feature similarity (Figure 3 (b)). Therefore, Attack B witnesses a downgrade in its performance when taking node features and their similarity into consideration.

## B.2 Existence of DSV for Other Protected Node Features

Table 11 reports the attack performance of link subgroups for Facebook and Spammer datasets while the subgroups are defined on other features instead of gender. Our main observation is that the attack has different performance across different link subgroups in all the settings, regardless of the selection of protected feature. For example, as shown in Table 11, when launching Attack B against GCN model on Facebook dataset, the BAA performance can reach 0.844 on the subgroup $G_0$ (C-C group), but drops to 0.697 on $G_2$ (NC-NC group), leading to DSV as large as 0.098 between these two groups. In other words, the links between the users with college degree are much more vulnerable to LMIA than those between the users with non-college degree. The similar observation also holds on F1-score metric. For example, when launching Attack B against GAT model on Pokec dataset, the F1-score can reach 0.782 on $G_0$, but drops to 0.658 on $G_2$, leading to DSV as 0.083. Such observations are consistent with the ones in Section 4.3.

Our second finding is that, the groups are always more vulnerable than the others regardless of the attack evaluation metric, the attack type, and the type of target GNN model. For example, the subgroup $G_0$, which is the one of the highest density always has the highest attack performance (BAA and F1-score), while the subgroup $G_2$, which is of the lowest density, always have the lowest attack performance. This implies that graph structure plays an important role in the existence of DSV—some subgroups are inherently more vulnerable to the attacks than the others due to their graph structure. These observations are also consistent with the ones in Section 4.3.

## B.3 Pearson Correlation and Causal Effect of Attack A

Table 12 presents the Pearson correlation between the five graph properties (i.e., group size, assortativity, density, AEBC, ANS and the attack performance of Attack A. We observe a weak Pearson correlation between the group size and attack performance as well as between the assortativity and attack performance. Thus both group size and assortativity do not impact attack performance much. On the other hand, we observe strong correlations between density/AEBC/ANS and attack performance. Among these three properties, both density and AEBC are negatively correlated with attack accuracy, while ANS is positively correlated with attack accuracy. In particular, the absolute values of the correlations always exceed 0.5 in all the settings and can be as high as 0.77, 0.63, and 0.66 for density, AEBC, and ANS respectively. These observations are consistent with the ones of Attack B (Section 5.2).

Table 13 presents the ATE of density/AEBC/ANS on attack performance of Attack A. The observations are similar to the results of Attack B (Section 5.2); thus we omit the discussions for simplicity.

## B.4 Explanations of Correlation between Group Density and Attack Performance

Intuitively, the embedding of a node aggregates more (less) information from its neighbors in a dense (sparse) group. Thus, it is more (less) challenging for LMIA to predict particular neighbors of a node from its embedding (and the target model output) if the node belongs to a dense (sparse) group. If this holds, then it can explain why a dense (sparse) group has a higher (lower) attack performance. To explain this reasoning quantitatively, we measure the similarity between nodes of member and non-member links. For demonstration, we focus on the three link subgroups in Spammer dataset, where the node similarity is measured as the Euclidean distance between the nodes' posterior output by the target model. Figure 4 presents the distribution of node similarity for member and non-member links in different link subgroups, which are sorted by their density in ascending order. From the results, we can observe that member and non-member links in $G_0$ (i.e., the sparsest group) are more distinguishable by their node similarity than those in $G_1$ and $G_2$. This explains why $G_0$ is the most vulnerable group among the three subgroups. For $G_2$ (i.e., the densest group), on the other hand, its node similarity on members and non-members are close to each other. Therefore, the members and non-members in $G_2$ are less distinguishable by their node similarity, leading to the least vulnerability among the three subgroups.

## B.5 Explanations of Correlation between AEBC and Attack Performance

To answer the question of why the attack performance negatively correlated with AEBC, we measure the difference $diff(G)$ in the dissimilarity of node pairs on member and non-member edges for each sampled subgroup. Formally, for each group $G$, we calculate

**Table 11: Disparity in subgroup vulnerability. $G_0$, $G_1$ and $G_2$ are link subgroups that are sorted by the group density in ascending order. The most and least vulnerable groups (i.e., the groups of the highest and lowest balanced attack accuracy respectively) are marked with green and pink respectively.**

**(a) Balanced Attack Accuracy (BAA)**

| Settings | | GAT | | | | GCN | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $G_0$ | $G_1$ | $G_2$ | **DSV** | $G_0$ | $G_1$ | $G_2$ | **DSV** |
| Facebook Education Level | Attack A | 0.811 | 0.763 | 0.693 | 0.079 | 0.854 | 0.827 | 0.689 | 0.092 |
| | Attack B | 0.827 | 0.779 | 0.718 | 0.073 | 0.844 | 0.812 | 0.697 | 0.098 |
| Pokec Marital Status | Attack A | 0.729 | 0.717 | 0.651 | 0.052 | 0.708 | 0.655 | 0.629 | 0.053 |
| | Attack B | 0.814 | 0.767 | 0.695 | 0.052 | 0.821 | 0.784 | 0.722 | 0.066 |

**(b) Attack F1-score**

| Settings | | GAT | | | | GCN | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $G_0$ | $G_1$ | $G_2$ | **DSV** | $G_0$ | $G_1$ | $G_2$ | **DSV** |
| Facebook Education Level | Attack A | 0.783 | 0.722 | 0.691 | 0.061 | 0.790 | 0.728 | 0.696 | 0.063 |
| | Attack B | 0.799 | 0.743 | 0.709 | 0.060 | 0.814 | 0.799 | 0.707 | 0.071 |
| Pokec Marital Status | Attack A | 0.741 | 0.721 | 0.637 | 0.069 | 0.729 | 0.714 | 0.653 | 0.051 |
| | Attack B | 0.782 | 0.736 | 0.658 | 0.083 | 0.777 | 0.709 | 0.674 | 0.069 |

**Table 12: Pearson correlation between attack performance and structural properties (Attack A). GD: group density; AEBC: average group betweenness centrality; ANS: average group node similarity**

| Dataset | GNN | Balanced attack accuracy (BAA) | | | | | Attack F1-score | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Group size** | **Assortativity** | **GD** | **AEBC** | **ANS** | **Group size** | **Assortativity** | **GD** | **AEBC** | **ANS** |
| Facebook | GAT | -0.05 | -0.10 | -0.61 | -0.51 | 0.60 | 0.03 | -0.03 | -0.74 | -0.61 | 0.63 |
| | GCN | 0.07 | -0.01 | -0.66 | -0.57 | 0.58 | 0.07 | -0.12 | -0.67 | -0.52 | 0.66 |
| Pokec | GAT | 0.03 | -0.08 | -0.71 | -0.62 | 0.63 | -0.09 | -0.05 | -0.72 | -0.59 | 0.61 |
| | GCN | 0.03 | -0.10 | -0.65 | -0.52 | 0.59 | 0.02 | -0.06 | -0.66 | -0.61 | 0.64 |
| Spammer | GAT | 0.04 | -0.03 | -0.77 | -0.53 | 0.55 | -0.04 | -0.07 | -0.70 | -0.61 | 0.63 |
| | GCN | -0.02 | -0.12 | -0.68 | -0.57 | 0.61 | -0.01 | 0.03 | -0.65 | -0.57 | 0.62 |

**Table 13: Counterfactual causal effect of structural properties on attack performance of Attack A (GD: group density; AEBC: average group betweenness centrality; ANS: average group node similarity).**

| Dataset | GNN | Balanced attack accuracy (BAA) | | | Attack F1-score | | |
|---|---|---|---|---|---|---|---|
| | | **GD** | **AEBC** | **ANS** | **GD** | **AEBC** | **ANS** |
| Facebook | GAT | -0.64 | -0.18 | 0.37 | -0.48 | -0.22 | 0.35 |
| | GCN | -0.49 | -0.12 | 0.33 | -0.47 | -0.19 | 0.23 |
| Pokec | GAT | -0.42 | -0.20 | 0.29 | -0.41 | -0.17 | 0.31 |
| | GCN | -0.34 | -0.09 | 0.26 | -0.37 | -0.19 | 0.34 |
| Spammer | GAT | -0.50 | -0.17 | 0.38 | -0.53 | -0.12 | 0.21 |
| | GCN | -0.49 | -0.19 | 0.33 | -0.46 | -0.11 | 0.24 |

$diff(G)$ as:

$$diff(G) = \frac{\sum_{e(u,v)\in G^-} d(f(u), f(v))}{|G^-|} - \frac{\sum_{e(u,v)\in G^+} d(f(u), f(v))}{|G^+|} \tag{16}$$

where $d()$ denotes a distance metric, $f()$ denotes the output of target model, $G^+$ and $G^-$ denote the set of the members and the non-members in $G$ respectively. In this paper we use Euclidean distance as the distance function $d()$ (i.e., $d(f(u) - f(v)) = ||f(u) - f(v)||_2$). Intuitively, higher $d(f(u), f(v))$ indicates that the posteriors of node $u$ and $v$ are more dissimilar. As the connected node pairs (i.e., members) have more similar posterior outputs to each other than the disconnected node pairs (i.e., non-members) [34], the attack should have higher accuracy on higher $diff()$ values as members and non-members are more distinguishable.

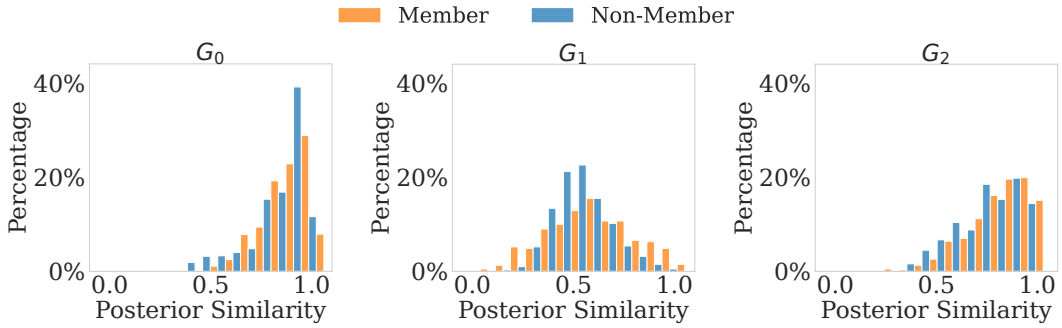



**Figure 4: Distribution of similarity of nodes on member and non-member links (Spammer dataset) for link subgroups. $G_0$ (M-M), $G_1$ (F-M) and $G_2$ (F-F) are the link groups sorted by their density in ascending order.**

**Table 14: Comparison of difference between posterior similarity of nodes on member and non-member edges for subgroups with top 50% high AEBC ("High AEBC" column) and bottom 50% AEBC ("Low AEBC" column).**

| Dataset | GAT | | GCN | |
|---|---|---|---|---|
| | High AEBC | Low AEBC | High AEBC | Low AEBC |
| Facebook | 0.218 | 0.257 | 0.010 | 0.044 |
| Pokec | 0.012 | 0.039 | 0.054 | 0.144 |
| Spammer | 0.064 | 0.085 | 0.082 | 0.132 |

**Table 15: Comparison of difference between posterior similarity of nodes on member and non-member edges for subgroups with top 50% high ANS ("High ANS" column) and bottom 50% ANS ("Low ANS" column).**

| Dataset | GAT | | GCN | |
|---|---|---|---|---|
| | High ANS | Low ANS | High ANS | Low ANS |
| Facebook | 0.255 | 0.214 | 0.062 | 0.014 |
| Pokec | 0.032 | 0.011 | 0.142 | 0.056 |
| Spammer | 0.081 | 0.067 | 0.113 | 0.099 |

Table 14 presents the average $diff(G)$ of the subgroups with top-50% lowest (low AEBC) and top-50% highest AEBC (high AEBC). We normalize the value of $diff(G)$ to be within the $[0, 1]$ range. From the results, we observe that the subgroups with high AEBC always have lower difference values than the subgroups with low AEBC: in all the settings, the difference values of the subgroups with low AEBC is 18%~340% higher than the subgroups with high AEBC. This suggests that the members in the subgroups of high AEBC are less distinguishable from the non-members, which leads to lower attacker performance. This observation explains the negative correlation between AEBC and attack performance on the subgroups.

## B.6 Explanations of Pearson Correlation between ANS and Attack Performance

To answer the question of why the attack performance is positively correlated with ANS, we follow the similar reasoning as in Appendix B.5. First, we measure the difference $diff(G)$ (Eqn. (16)) between the posterior similarity of node pairs on member and non-member edges. Next, we calculate the average of the difference values of the subgroups with top 50% high ANS and the subgroups with bottom 50% ANS.

Table 15 presents the results of the difference in node posterior similarity for subgroups of high and low ANS. We observe that the subgroups with higher ANS have higher different values than the subgroups with lower ANS: the difference in node posterior similarity between nodes on member and non-member edges for the subgroups with low AEBC are 14% ~343% higher than the subgroups with high AEBC. As higher difference leads to better attack performance, this observation explains the positive correlation between ANS and attack performance.

# C PROOF OF THEOREMS AND LEMMA

## C.1 Proof of Theorem 1

Proof. Given a graph $\mathcal{G}$ that has $k_m$ member links and $k_o$ non-member links, its density $d$ is measured as $d = \frac{k_m}{k_o+k_m}$. Similarly, the density $d_i$ of a link group $G_i \in \mathcal{G}$ is calculated as $d_i = \frac{k_m^i}{k_o^i+k_m^i}$. Following Equations (9) and Equation (11), the expected density $E(d_i)$ of $G_i$ in each forward process by FairDefense is:

$$\begin{aligned} E(d_i) &= \frac{k_m^i(1-p_1) + k_o^i p_1 \frac{k_m}{k_o}}{k_m^i + k_o^i} \\ &= d_i(1-p_1) + p_1(1-d_i)\frac{k_m}{k_o} \end{aligned} \tag{17}$$

For two subgroups $G_i$ and $G_j$, their densities are $d_i$ and $d_j$ respectively. Without loss of generality, we assume $d_i > d_j$. Formally,

the difference between their expected densities is:

$$\begin{aligned} E(d_i) - E(d_j) &= (1-p_1)(d_i - d_j) + p_1(d_j - d_i)\frac{k_m}{k_o} \\ &= (d_i - d_j)(1 - p_1 - p_1\frac{k_m}{k_o}) \\ &= (d_i - d_j)(1 - p_1\frac{k_m + k_o}{k_o}) \\ &= (d_i - d_j)(1 - \frac{k_o}{e^\gamma k_m + k_o} \times \frac{k_m + k_o}{k_o}) \\ &= (d_i - d_j)(1 - \frac{k_m + k_o}{e^\gamma k_m + k_o}) \end{aligned} \tag{18}$$

Considering $\gamma \geq 0$, $(1 - \frac{k_m+k_o}{e^\gamma k_m+k_o}) \geq 0$ is always satisfied. Thus we can add absolute signs to both sides of the equation above as:

$$|E(d_i) - E(d_j)| = |d_i - d_j|(1 - \frac{k_m + k_o}{e^\gamma k_m + k_o})$$

This completes the proof. □

## C.2 Proof of Lemma 2

Proof. Given a graph $\mathcal{G}$ that has $k_m$ member links and $k_o$ non-member links. For any node pair $(u, v) \in \mathcal{G}$, if $(u, v)$ is "seen" as a member (i.e., $A[u, v] = 1$), there will be two possible cases:

- $e(u, v)$ exists in the graph $\mathcal{G}$ and it is remained as a member.
- $e(u, v)$ does not exist in the graph $\mathcal{G}$ and it is changed to a member.

Formally, the probability of the first case $p(A[u, v] = 1, e(u, v) \in \mathcal{G})$ is calculated as:

$$\begin{aligned} &p(A[u,v] = 1, e(u,v) \in \mathcal{G}) \\ &\quad = p(A[u,v] = 1 | e(u,v) \in \mathcal{G}) \times p(e(u,v) \in \mathcal{G}) \\ &\quad = (1 - p_1) \times \frac{k_m}{k_m + k_o} \end{aligned} \tag{19}$$

Similarly, the probability of the second case $p(A[u, v] = 1, e(u, v) \notin \mathcal{G})$ is:

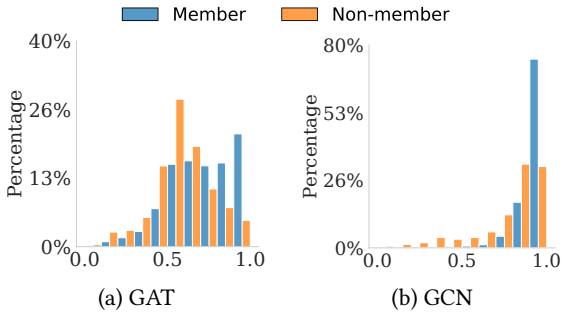
$$\begin{aligned} &p(A[u,v] = 1, e(u,v) \notin \mathcal{G}) \\ &\quad = p(A[u,v] = 1 | e(u,v) \notin \mathcal{G}) \times p(e(u,v) \notin \mathcal{G}) \\ &\quad = p_2 \times \frac{k_o}{k_m + k_o} \end{aligned} \tag{20}$$

Combining the probabilities above together, we have the probability $p_m$ that $(u, e)$ is "seen" as a member:

$$\begin{aligned} p_m &= p(A[u,v] = 1, e(u,v) \in \mathcal{G}) + p(A[u,v] = 1, e(u,v) \notin \mathcal{G}) \\ &= (1 - p_1) \times \frac{k_m}{k_m + k_o} + p_2 \times \frac{k_o}{k_m + k_o} \end{aligned} \tag{21}$$

Following the same logic, we can calculate the probability $p_n$ of $(u, v)$ to be "seen" as a non-member by adding up the probabilities of two cases:

- $e(u, v)$ exists in the graph $\mathcal{G}$ and it is changed to a non-member.
- $e(u, v)$ does not exist in the graph $\mathcal{G}$ and it is remained as a non-member.




(a) GAT (b) GCN

**Figure 5: Distribution of posterior similarity of nodes on member and non-member edges after the deployment of DP-SGD ($\epsilon = 0.1$, Spammer dataset).**

Similarly, the probability $p_n$ will be calculated as:

$$\begin{aligned} p_n &= p(A[u,v] = 0, e(u,v) \in \mathcal{G}) + p(A[u,v] = 0, e(u,v) \notin \mathcal{G}) \\ &= p_1 \times \frac{k_m}{k_m + k_o} + (1 - p_2) \times \frac{k_o}{k_m + k_o} \end{aligned} \tag{22}$$

According to the definition of $p_1$ and $p_2$ (Equations (9) & (11)), it is easy to prove that $p_m$ and $p_n$ are constants regardless of the value of $\gamma$: considering $p_1 : p_2 = k_o : k_m$, Eqn. (21) can be reformatted as:

$$\begin{aligned} p_m &= (1 - p_1) \times \frac{k_m}{k_m + k_o} + p_1 \times \frac{k_m}{k_m + k_o} \\ &= \frac{k_m}{k_m + k_o} \end{aligned} \tag{23}$$

In similar logic, Eqn. (22) can be reformatted as:

$$\begin{aligned} p_n &= p_2 \times \frac{k_o}{k_m + k_o} + (1 - p_2) \times \frac{k_o}{k_m + k_o} \\ &= \frac{k_o}{k_m + k_o} \end{aligned} \tag{24}$$

This completes the proof. □

## C.3 Proof of Theorem 3

For the sparse graphs (i.e., $k_m < k_o$), a node pair is less likely to be a member edge (i.e., connected) in the perturbed graph than as a non-member (i.e., $p_m < p_n$). On the other hand, for the dense graphs (i.e., $k_m > k_o$), a node pair is more likely to be a member edge (i.e., connected) in the perturbed graph than as a non-member (i.e., $p_m > p_n$). As the probability of an edge to be a member in the original training graph is also $\frac{k_m}{k_m+k_o}$, the perturbation by FairDefense does not change such probability (Eqn. (14)). We also have the same observation for the non-member edges.

# D ADDITIONAL RESULTS OF DEFENSE

## D.1 Why DP-SGD Is Not Effective against LMIA?

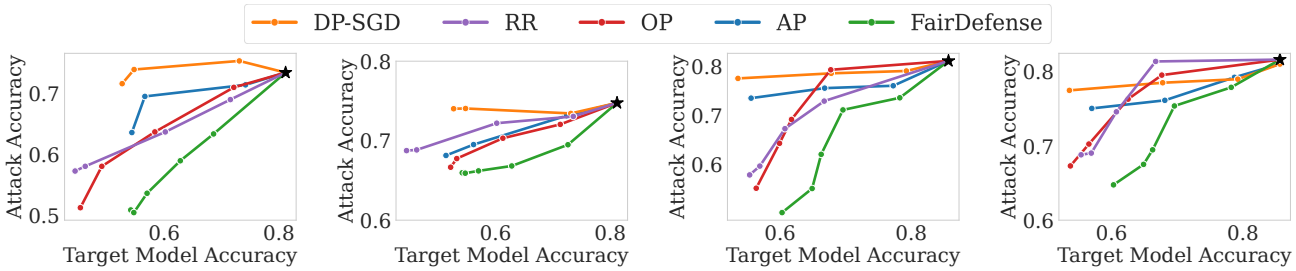
It is expected that DP-SGD should be effective in reducing LMIA accuracy. However, our results (Table 9) show that DP-SGD barely affects LMIA performance, even with strong noise. For instance, balanced attack accuracy drops only 5% for Attack A against GAT

**Table 16: DSV mitigation and defense effectiveness of FairDefense and four baselines (Spammer dataset, BAA as the attack metric, Attack B). The lowest MIA accuracy and DSV after mitigation is marked with pink and green respectively. $G_0$, $G_1$, and $G_2$ are link groups (F-F, M-F, M-M) that are sorted by group density in ascending order.**

| Settings | | GAT | | | GCN | | |
|---|---|---|---|---|---|---|---|
| | | Target Acc. | Attack Acc. | DSV | Target Acc. | Attack Acc. | DSV |
| Spammer | No defense | 0.681 | 0.684 | 0.101 | 0.712 | 0.589 | 0.069 |
| | DP-SGD | 0.561 | 0.685 | 0.108 | 0.569 | 0.591 | 0.053 |
| | RR | 0.558 | 0.522 | 0.086 | 0.571 | 0.478 | 0.071 |
| | OP | 0.552 | 0.514 | 0.045 | 0.581 | 0.513 | 0.047 |
| | AP | 0.565 | 0.625 | 0.089 | 0.562 | 0.559 | 0.044 |
| | FairDefense | 0.563 | 0.470 | 0.034 | 0.572 | 0.461 | 0.038 |

**Table 17: DSV mitigation and defense effectiveness of FairDefense and four baselines (Pokec dataset, BAA as the attack metric, Attack A). The lowest MIA accuracy and DSV after mitigation is marked with pink and green respectively. $G_0$, $G_1$, and $G_2$ are link groups (F-F, M-F, M-M) that are sorted by group density in ascending order.**

| Settings | | GAT | | | GCN | | |
|---|---|---|---|---|---|---|---|
| | | Target Acc. | Attack Acc. | DSV | Target Acc. | Attack Acc. | DSV |
| Pokec | Original | 0.657 | 0.683 | 0.064 | 0.687 | 0.627 | 0.073 |
| | DP-SGD | 0.594 | 0.687 | 0.059 | 0.592 | 0.666 | 0.068 |
| | RR | 0.600 | 0.634 | 0.057 | 0.593 | 0.575 | 0.067 |
| | OP | 0.591 | 0.538 | 0.042 | 0.600 | 0.549 | 0.037 |
| | AP | 0.588 | 0.624 | 0.052 | 0.581 | 0.619 | 0.041 |
| | FairDefense | 0.592 | 0.497 | 0.037 | 0.596 | 0.508 | 0.018 |




(a) Attack A on GAT (b) Attack B on GAT (c) Attack A on GCN (d) Attack B on GCN

**Figure 6: Trade-off between defense and target model accuracy (Facebook dataset, BAA as the attack accuracy metric). The ⋆ mark denotes no defense. The defense with lower attack accuracy and higher target model accuracy has a better trade-off.**

on the Spammer dataset. To investigate the reason, we analyze how the similarity between the nodes on members and non-member links is affected by DP-SGD. In particular, we randomly sampled 10,000 links (members) and 10,000 disconnected node pairs (non-members) from the Spammer dataset, and measure the similarity between the node pairs on these member/non-member links as the normalized L2 distance between the posteriors of these two nodes.

Figure 5 presents the results of similarity distribution for both GAT and GCN models. We observe that there still exists a significant gap between the similarity of node pairs on member and non-member links. In other words, members and non-members are still well distinguishable. This can be explained as follows: as DP-SGD adds noise to the gradients during training, it only changes the model parameters but has no direct influence on the graph structure. Hence, DP-SGD fails to provide an effective defense against the inference attack.

## D.2 Performance of FairDefense on Attack B

Table 16 presents the results of defense effectiveness and DSV mitigation for Attack B on both GAT and GCN models and the Spammer dataset. Similar to the result of Attack A (§6.2), FairDefense is effective in both defense and DSV mitigation. Furthermore, FairDefense outperforms the baselines in both attack accuracy and DSV.

## D.3 Performance of FairDefense on Facebook and Pokec Datasets

Table 17 presents the results of defense effectiveness and DSV mitigation for Attack A against both GAT and GCN on Pokec dataset. We observe similar patterns as Spammer dataset (§6.2) — FairDefense shows its defense effectiveness in these settings too. Furthermore, FairDefense outperforms the baselines in defense effectiveness and DSV mitigation in all the settings.

In Figure 6 we present the trade-off between defense and target model accuracy on Facebook dataset. We observe similar patterns as the Spammer dataset (§6.2) — FairDefense better addresses the trade-off between defense and target model performance than the four baselines.