Demystifying Uneven Vulnerability of Link Stealing Attacks against Graph Neural Networks

He Zhang [1] **Bang Wu** [1] **Shuo Wang** [2] **Xiangwen Yang** [1] **Minhui Xue** [2] **Shirui Pan** [3] **Xingliang Yuan** [1]

Abstract

While graph neural networks (GNNs) dominate the state-of-the-art for exploring graphs in real-world applications, they have been shown to be vulnerable to a growing number of privacy attacks. For instance, link stealing is a well-known membership inference attack (MIA) on edges that infers the presence of an edge in a GNN's training graph. Recent studies on independent and identically distributed data (e.g., images) have empirically demonstrated that individuals from different groups suffer from different levels of privacy risks to MIAs, i.e., uneven vulnerability. However, theoretical evidence for such uneven vulnerability is missing. In this paper, we first present theoretical evidence of the uneven vulnerability of GNNs to link stealing attacks, which lays the foundation for demystifying such uneven risks among different groups of edges. We further demonstrate a group-based attack paradigm to expose the practical privacy harm to GNN users derived from the uneven vulnerability of edges. Finally, we empirically validate the existence of obvious uneven vulnerability on ten real-world datasets (e.g., about 25% AUC difference between different groups in the Credit graph). Compared with existing methods, the outperformance of our group-based attack paradigm confirms that customising different strategies for different groups results in more effective privacy attacks.

[1]Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Australia [2]CSIRO's Data61, Australia [3]School of Information and Communication Technology, Griffith University, Australia. Correspondence to: Xingliang Yuan <xingliang.yuan@monash.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Graphs depict complex data from a wide range of real-world applications due to their flexibility in representing objects and their interactions (Wang et al., 2022; Galmés et al., 2022; Luo et al., 2022). Taking a graph in social networks (Backstrom & Leskovec, 2011) for example, nodes represent individuals and edges indicate connection (e.g., friendship) between individuals. With the advancement of graph neural networks (GNNs) (Zheng et al., 2022c; Jin et al., 2022a; Zheng et al., 2022d; Jin et al., 2022b; Liu et al., 2023b), they are now widely deployed in practise as a method to explore graph data. For example, GNNs can provide personalised searches and recommendations to users on social media or doctor recommendation systems, dramatically enriching people's daily life (Pal et al., 2020).

Along with their widespread deployment, GNNs are increasingly becoming the target of privacy attacks due to the private information contained in them (e.g., graph data, model architecture and parameters (Duddu et al., 2020; Wu et al., 2022a)), whose leakage can benefit privacy attackers in various aspects (e.g., reducing effort in collecting graph data). However, users of GNN systems desire to keep their personal information private, especially for sensitive applications. For example, in a GNN-based doctor recommendation system that connects patients and specialist doctors (Mondal et al., 2020), information leakage between a patient and a heart specialist indicates that attackers can infer whether the patient has heart disease, consequently leading to trust crises in GNN systems. Therefore, there is an urgent need to fully reveal the privacy risks of GNNs.

As a typical privacy attack on GNNs, membership inference attacks (MIAs) attempt to infer if a particular sample or graph component exists in the training dataset. In addition to determining the membership of nodes/graphs (He et al., 2021b; Wu et al., 2021a; Wang & Sun, 2021), attackers can also launch MIAs targeting edges. According to a recent method called *link stealing* (He et al., 2021a), attackers can infer the connection between any two nodes in the training graph based solely on their prediction distributions. It exposes the privacy risk of edges in the most practical setting when deploying GNNs, where attackers only need to query target GNNs (i.e., a black-box setting) to launch attacks.

Currently, almost all MIA methods are designed to be optimal in terms of the overall population with a shared discriminator. However, this attack paradigm underestimates the privacy risk of certain groups. Existing studies (Chang & Shokri, 2021; Kulynych et al., 2022; Zhong et al., 2022), which focus on independent and identically distributed (IID) data (e.g., image data (Wu et al., 2022c)), have empirically demonstrated that a model's vulnerability to MIAs differs across groups, i.e., privacy risk is not uniformly distributed (Kulynych et al., 2022). Given these studies, one research question (RQ) concerning the privacy of edges in training graph of GNNs is **(RQ 1)** "*Do edges in the training graph of GNNs have an uneven vulnerability?*" If yes, what are the potential factors responsible for this uneven vulnerability? Furthermore, if an uneven vulnerability exists, GNN users are interested in knowing whether it will practically harm their privacy. Thus, another research question is **(RQ 2)** "*Could intelligent attackers who are aware of this uneven vulnerability benefit from it?*"

To answer these questions, we need to dive into the uneven vulnerability concerning edge privacy in GNNs. However, it is not trivial to directly adapt current studies to link stealing attacks in GNNs for the following reasons. First, to the best of our knowledge, there is no theoretical evidence regarding uneven vulnerability to MIAs, when the current literature is largely empirical in nature. Second, in link stealing attacks, privacy leakage is associated with the connections between individuals, rather than unconnected individual samples as in previous studies on IID data. Finally, edges can be stolen from GNNs that are not designed for link prediction tasks, implying that the rationale of link stealing is different from general MIAs against deep neural networks.

To this end, we model link stealing attacks with a hypothesis test and analyse the uneven vulnerability of GNNs on edges to answer RQ1. For RQ2, we propose a group-based attack paradigm to advance current link stealing attacks. The contributions of our paper are summarised as follows:

- We introduce the first theoretical analysis of uneven vulnerability in the context of GNNs, and pinpoint intra-class and inter-class node pairs [1] have different levels of vulnerability to link stealing attacks.
- We propose a group-based attack paradigm that employs customised strategies for different groups to launch attacks, which demonstrates the practical privacy harm derived from uneven vulnerability.
- We empirically evaluated uneven vulnerability on ten real-world datasets from different domains and presented the advantages of our group-based paradigm compared to generic link stealing attacks.

[1] In an intra/inter-class node pair, nodes have same/different predicted labels.

2. Related Work

In this section, we review current MIAs targeting GNNs and studies regarding uneven vulnerability to MIAs.

2.1. MIAs on GNNs

GNNs are neural network architectures developed to learn graph embedding for exploring graph data and accomplishing various graph-related tasks (Gilmer et al., 2017; Pan et al., 2020; Wan et al., 2021; Liu et al., 2023c; Tan et al., 2023; Liu et al., 2023a). Currently, many GNNs perform well by using the message passing mechanism (Zheng et al., 2022b), where the edges are involved in supporting the interaction between nodes (Wu et al., 2021b). However, recent works (Zhang et al., 2021; 2022b; 2023) have demonstrated that trustworthiness related issues exist in current GNNs (Zhang et al., 2022a). For example, targeting the privacy of victim models, MIAs aim to determine the existence of a sample in the training dataset of the victim models. According to the target type of inference, current MIAs on GNNs can be categorised into node-level, link-level, and graph-level attacks (Zhang et al., 2022a). Link-level MIAs (He et al., 2021a; Wu et al., 2022b) aim to infer if a specific edge/link is in the training graph of a victim GNN, even if the target GNN is not designed for the link prediction task.

Link stealing attacks are typical link-level MIAs against GNNs. According to the attackers' knowledge, He et al. (2021a) propose a thorough taxonomy of the threat model. The most realistic scenario is that the adversary has black-box access to a victim GNN model, where the only attack knowledge is the posteriors of nodes from target GNNs. Link stealing attacks are based on the intuition that two nodes that share similar attributes and/or predictions are more likely to be linked, i.e., homophily. Therefore, He et al. (2021a) propose that attackers can utilise the distance between two nodes to infer the existence of an edge in this pair of nodes. However, current link stealing attacks treat all node pairs equally and use a shared single threshold to infer edges, which potentially underestimates the privacy risk of connection status in node pairs from some groups.

2.2. Theoretical Study on MIAs

The first MIA on machine learning models is proposed by Shokri et al. (2017), who also show how it relates to model overfitting. Farokhi & Kâafar (2020) evaluate the amount of information leakage from the victim model under MIAs through conditional mutual information leakage. Although some studies show that the vulnerability to MIAs is bounded by differential privacy (Yeom et al., 2018; Cherubin et al., 2019), Humphries et al. (2020) found that these bounds only hold when the victim model owns independent and identically distributed training data. Jayaraman et al. (2021)

propose that attackers can infer private training data based on hypothesis testing on the output of differential private mechanisms. Murakonda et al. (2021) analyse the privacy risk bound of graphical models via employing hypothesis testing and likelihood ratio test. However, the risk bound (Murakonda et al., 2021) cannot be applied to link stealing attacks since GNNs are different from graphical models. Following these works, in this paper, we employ the hypothesis test to model link stealing attacks.

2.3. Uneven Vulnerability

The model disparity in the field of machine learning has historically focused on whether a model can tolerate individual differences and offer the same level of service (e.g., accuracy) to individuals with different backgrounds (Saxena et al., 2019; Mehrabi et al., 2021). Chang & Shokri (2021) empirically demonstrate that individuals from different groups have different privacy risks. Moreover, Kulynych et al. (2022) characterise the vulnerability to MIAs and show an analysis of the uneven vulnerability. However, applying these empirical observations to link stealing attacks is non-trivial since GNNs memorise the edge differently and implicitly. Furthermore, there is no theoretical evidence to support the uneven vulnerability in existing studies.

3. Problem Formulation

This section first introduces the victim GNN model (e.g., training data, model architecture and task), followed by representing the adversary model of link stealing attacks.

3.1. Victim GNN Model

Graphs. A graph $G = \{\mathcal{V}, \mathcal{E}\}$ consists of a node set $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$ and edge set $\mathcal{E}$. $\mathcal{E}$ characterises the relationship information in G. The edge set can also be denoted by an adjacency matrix $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathbf{A}_{i,j} = 1$ when $e_{ij} = (v_i, v_j) \in \mathcal{E}$, otherwise $\mathbf{A}_{i,j} = 0$. Node features are denoted by $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times k}$ (k indicates the dimensionality of features), and the i-th row of $\mathbf{X}$ indicates the feature of node v_i. Without loss of generality, another description form of a graph is $G = \{\mathbf{A}, \mathbf{X}\}$. In this paper, we focus on the undirected graph, that is, $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$.

GNNs. In this paper, we focus on common message-passing-based GNNs, where the node embedding is repeatedly updated by stacking operations as follows (Gilmer et al., 2017):

$$\begin{aligned} \mathbf{m}_v^{(t)} &= \sum_{u\in\mathcal{N}(v)} M_t(\mathbf{h}_u^{(t-1)}, \mathbf{h}_v^{(t-1)}), \\ \mathbf{h}_v^{(t)} &= U_t(\mathbf{h}_v^{(t-1)}, \mathbf{m}_v^{(t)}), \end{aligned} \tag{1}$$

where $\mathbf{h}_v^{(t)}$ indicates the node embedding of v at layer $t \in \{1, \ldots, T\}$, and the neighbour node set of v is denoted by $\mathcal{N}(v)$. At layer t, $M_t(\cdot,\cdot)$ indicates the message function and $U_t(\cdot,\cdot)$ represents the embedding updating function.

Task. The victim GNNs are designed to perform node classification. For a graph $G = \{\mathcal{V}, \mathcal{E}\}$, the set of labelled nodes is denoted by $\mathcal{V}_l \subset \mathcal{V}$, where y_i is the label of $v_i \in \mathcal{V}_l$. The set of unlabelled nodes in G is indicated by $\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_l$. Given G and node labels, node classification aims to train a GNN model f, which can predict labels for nodes in $\mathcal{V}_u$.

3.2. Threat Model

Attacker's Goal and Capacity. In this paper, we focus on a popular attack called link stealing (He et al., 2021a). This attack aims to steal the training graph structure, which is typically considered confidential information held by model developers, and often contains sensitive information (He et al., 2021a; Wu et al., 2022b). Specifically, the attacker, denoted as $\mathcal{A}$, will infer whether there exists an edge between any two nodes in the training graph G of a victim GNN f during its inference. We assume that this attacker $\mathcal{A}$ has black-box access to the victim GNN f. Namely, the attacker can access neither the model parameters nor its internal representation during inference. Instead, they can only issue queries from any node v to f, and obtain the probability distribution of label prediction (i.e., $f(v)$). This scenario presents the greatest level of challenge yet remains a realistic circumstance for adversaries in numerous real-world applications (He et al., 2021a). For example, for a GNN deployed in machine learning as a service (MLaaS), service providers (e.g., Amazon (Adeshina, 2020), Google (Lackey, 2022)) only provide model prediction APIs to customers during the runtime of model serving due to security and privacy concerns. Thus, the attacker exploiting these APIs can only issue queries and obtain responses from f.

Link Stealing Attacks for GNNs. As shown in Eq. (1), the embedding of v is updated by aggregating the embedding of nodes in $\mathcal{N}(v)$, so the connected nodes are prone to have similar prediction results. Inspired by this, the attacker $\mathcal{A}$ uses the distance $d(f(v_i), f(v_j))$ to infer whether a specific edge $e_{ij} = (v_i, v_j)$ is in the edge set $\mathcal{E}$ of graph $G = \{\mathcal{V}, \mathcal{E}\}$, which is the training graph of victim GNN f. Given $d(f(v_i), f(v_j)) \in [0,1]$ and $d(f(v_i), f(v_j)) = 0$ only when $f(v_i) = f(v_j)$, the link prediction score is calculated as $s(i,j) = 1 - d(f(v_i), f(v_j))$. Let $\tau \in [0,1]$ be the threshold for discriminating the existence of $e_{ij} = (v_i, v_j)$ in $\mathcal{E}$, the results $\mathbf{A}^{pred}$ of link stealing attacks is

$$\mathbf{A}_{i,j}^{pred} = \begin{cases} 1 & \text{if } s(i,j) \geq \tau \\ 0 & \text{if } s(i,j) < \tau. \end{cases} \tag{2}$$

Remarks. Current link stealing attacks (He et al., 2021a) treat all node pairs equally. However, existing studies (Chang & Shokri, 2021; Kulynych et al., 2022) show that the privacy risk to MIAs is not evenly distributed. Although

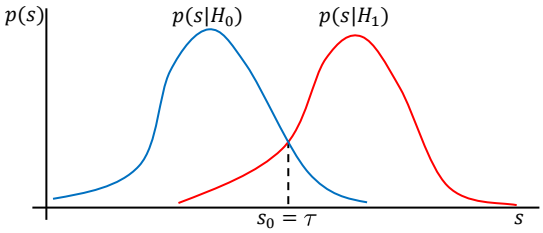

Figure 1. Distributions of $p(s|H_0)$ and $p(s|H_1)$ and the optimal decision boundary τ in statistical decision (Duda et al., 2001).

some works analyse privacy risks to MIAs, there is no theoretical evidence supporting the existence of uneven vulnerability. Our paper substantially differs from these studies in that: (1) we theoretically demonstrate the existence of the uneven vulnerability in link stealing attacks and empirically validate it; (2) we propose that intelligent attackers can excavate the knowledge they own and utilise the uneven vulnerability for devising group-based link stealing attacks.

4. Group-based Attacks Driven by Uneven Vulnerability

4.1. Overview

The intuition of group-based attacks is that node pairs from different groups have different levels of vulnerability. Concretely, for a node pair (v_i, v_j), the attacker $\mathcal{A}$ can use their label predictions $f(v_i)$ and $f(v_j)$ to categorise this pair into intra-class or inter-class groups. Consequently, $\mathcal{A}$ can employ customised attack strategies for each group to carry out more effective attacks, since using a single strategy is suboptimal. In this section, we first present how to measure the vulnerability to MIAs in 4.2. Based on this metric, we provide theoretical evidence supporting the uneven vulnerability across groups in section 4.3. Finally, in Section 4.4, we propose that attackers can devise a group-based method to launch more effective link stealing attacks.

4.2. Metric of Vulnerability to MIAs

In this paper, we use the hypothesis test to evaluate the privacy risk of edges in the context of link stealing attacks.

Hypothesis Test. In membership inference attacks on a given pair of nodes (v_i, v_j), the attacker $\mathcal{A}$ aims to discriminate the following two hypotheses:

- Null hypothesis H_0: In the training graph G, the edge $e_{ij} = (v_i, v_j) \notin \mathcal{E}$. $f(v_i)$ and $f(v_j)$ are predictions of the GNN model f trained on G with $\mathbf{A}_{i,j} = 0$.
- Alternative hypothesis H_1: In the training graph G, edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$. $f(v_i)$ and $f(v_j)$ are predictions of the GNN model f trained on G with $\mathbf{A}_{i,j} = 1$.

Given these two hypotheses H_0 and H_1, we use the quote symbol to denote the decision of attacker $\mathcal{A}$, i.e., "H_0" and "H_1". According to ground truth $\mathbf{A}_{i,j}$ and prediction $\mathbf{A}_{i,j}^{pred}$, attackers can potentially make two different errors and the probability of errors are denoted by $\Pr(\text{“}H_1\text{”} \mid H_0)$ and $\Pr(\text{“}H_0\text{”} \mid H_1)$, where $\Pr(\cdot)$ indicates the probability function. Following previous research (Murakonda et al., 2021), we use the error parameter α (i.e., false positive rate) and the power parameter β (i.e., true positive rate) to quantify the privacy risks of the victim GNN f. The error α measures $P(\text{“}H_1\text{”} \mid H_0)$, and power β measures $P(\text{“}H_1\text{”} \mid H_1)$.

Vulnerability Metric. In current MIAs on edges, attackers employ $s(i, j)$ as the link prediction score on node pair (v_i, v_j). Based on current studies on link prediction (Li et al., 2021) and node classification (Pan et al., 2018), we assume that $s(i, j) \sim N(\mu_0^s, \sigma^s)$ when $\mathbf{A}_{i,j} = 0$, $s(i, j) \sim N(\mu_1^s, \sigma^s)$ when $\mathbf{A}_{i,j} = 1$, and $\mu_1^s \geq \mu_0^s$, where $\mu_{0/1}^s$ and σ^s represent the mean and variance of link prediction scores. According to the *Neyman and Pearson lemma* (Neyman & Pearson, 1933), for a given α, the likelihood ratio test has the maximum β in all decision rules, i.e., the optimal threshold τ is determined by the interaction position of $N(\mu_0^s, \sigma^s)$ and $N(\mu_1^s, \sigma^s)$. As illustrated in Fig. 1, $z_{1-\alpha} + z_\beta$ measures the privacy risk of edges in the context of link stealing attacks, where $z_{1-\alpha}$ and z_β indicate the $1-\alpha$ and β quantile of $N(0, 1)$, respectively. In this paper, the vulnerability of a model f to MIAs is denoted by

$$V(f) = z_{1-\alpha} + z_\beta \tag{3}$$

4.3. Theoretical Evidence of Uneven Vulnerability

In this section, we aim to answer **RQ 1**, i.e., "*Do edges in the training graph of GNNs have an uneven vulnerability?*". Concretely, we first show how attackers can divide target node pairs into intra-class and inter-class groups according to their predicted labels. Then we define uneven vulnerability and prove that intra-class and inter-class node pairs have different levels of vulnerability to link stealing attacks.

Intra-class and Inter-class Groups. We assume that the victim GNN f is well trained for binary node classification for simplicity, which does not affect the generalisation of the following analysis to multi-class tasks. The label set of nodes in G is $\{y_0, y_1\}$. Here, we use $\mathbf{X}^{y_0}$ and $\mathbf{X}^{y_1}$ to denote the node features/embedding associated y_0 and y_1, respectively. Given $\mathbf{m} \in \{0, 1\}^N$ ($\mathbf{m}_i = 1$ indicates v_i belongs to class y_1, and otherwise $\mathbf{m}_i = 0$), we have $\mathbf{X}^{y_1} = diag(\mathbf{m})\mathbf{X}$, $\mathbf{X}^{y_0} = (\mathbf{I} - diag(\mathbf{m}))\mathbf{X}$, and $\mathbf{X} = \mathbf{X}^{y_0} + \mathbf{X}^{y_1}$. In this paper, we use $S(\cdot)$ to denote the grouping information of nodes and assume $S(v_i) = \mathbf{m}_i$ due to the attacker $\mathcal{A}$ being able to obtain $\mathbf{m}$ by querying the well-trained f. According to the grouping information of any two nodes, the linking situation between them can be categorised into $g_0^0 = \{\mathbf{A}_{i,j} = 0 \mid S(v_i) \neq S(v_j)\}$, $g_0^1 = \{\mathbf{A}_{i,j} = 1 \mid$

$S(v_i) \neq S(v_j)\}$, $g_1^0 = \{\mathbf{A}_{i,j} = 0 \mid S(v_i) = S(v_j)\}$ and $g_1^1 = \{\mathbf{A}_{i,j} = 1 \mid S(v_i) = S(v_j)\}$, and we have $g_0 = g_0^0 \cup g_0^1 = \{\mathbf{A}_{i,j} \mid S(v_i) \neq S(v_j)\}$ and $g_1 = g_1^0 \cup g_1^1 = \{\mathbf{A}_{i,j} \mid S(v_i) = S(v_j)\}$.

Uneven Vulnerability As introduced in Section 2.3, edges in different groups may own different levels of privacy risks. Since the likelihood test provides an upper bound for privacy risks to MIAs, the group-based vulnerability difference is defined as follows. Given $V(f) = z_{1-\alpha} + z_\beta$ and two groups of attack targets, i.e., $g_\triangleright$ and $g_\triangleleft$, the *vulnerability difference* between $g_\triangleright$ and $g_\triangleleft$ is defined as

$$\Delta V_{g_\triangleright, g_\triangleleft}(f) = \mid V_{g_\triangleright}(f) - V_{g_\triangleleft}(f) \mid, \tag{4}$$

where $|\cdot|$ calculates the norm and V_g is the vulnerability of model f on group g.

By calculating $\Delta V_{g_\triangleright, g_\triangleleft}(f)$ in the embedding space rather than the scoring space, we can evaluate how an edge's existence impacts the learnt embedding during message passing (i.e., Eq. (1)), where edges are involved in GNN architectures. For simplicity, we consider left normalisation $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}$, where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-connection, $\widetilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$ ($d_i = \mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$) is the degree matrix of $\widetilde{\mathbf{A}}$. Based on existing studies on graph learning (Pan et al., 2018; Li et al., 2021), we assume that $\mathbf{X}[t]^{y_i} \sim N(\mu_i^e, \sigma^e)$, where μ_i^e and σ^e indicate the mean and variance of the embedding of nodes from class y_i ($i = 0, 1$) at t-th layer. The following theorem shows that there exists a vulnerability difference between the node pair groups g_0 and g_1, i.e., $\Delta V_{g_0, g_1}(f) \neq 0$.

Theorem 4.1. *Given a well-trained GNN model f designed for node classification, inter-class node pairs (i.e., g_0) and intra-class node pairs (i.e., g_1) have different degrees of vulnerability to link stealing attacks, i.e.,*

$$\mathbb{E}\left[\Delta V_{g_0, g_1}(f)\right] = \frac{|(\mu_1^e - \mu_0^e)|}{\sigma^e} \left|\mathbb{E}\left[\delta_{g_0}\right] - \mathbb{E}\left[\delta_{g_1}\right]\right| \neq 0, \tag{5}$$

where δ is a variable related to node degree and homophily.

Proof. See Appendix A for details. □

In link stealing attacks (He et al., 2021a), attackers employ $s(i,j) = 1 - d(f(v_i), f(v_j))$ as the link prediction score between v_i and v_j. In a well-trained GNN f, given that $d_g(\cdot,\cdot)$ represents the distance function in the embedding space, g_0 and g_1 have different degrees of privacy risks since $d(f(v_i), f(v_j)) \propto d_g(v_i, v_j)$ and $\mathbb{E}\left[\Delta V_{g_0, g_1}(f)\right] \neq 0$. In addition to demonstrating the uneven vulnerability of edges across different groups, Theorem 4.1 also explains the trade-off between GNN performance and edge privacy (more discussions can be found in Appendix A).

Remarks. (1) We choose the *Gaussian distribution* in this paper for two main considerations. First, the Gaussian distribution is concise for theoretical analysis and widely used in existing graph learning methods (He et al., 2015; Egilmez et al., 2017; Bojchevski & Günnemann, 2018). Second, the Gaussian distribution follows the maximum entropy principle (Li & Liu, 2007), enabling it to be suitable for black-box attacks in which attackers have limited knowledge and want to maximise their attack benefits. (2) The analysis presented in Theorem 4.1 adheres to the convention in graph learning studies, which involves starting from a 1 layer operation (Kipf & Welling, 2017). It focusses on the most fundamental message-passing mechanisms of GNNs (Chen et al., 2023; Wu et al., 2021b), making it applicable to almost any GNN layer due to the stacking layer design of current GNNs (e.g., conducting Theorem 4.1 on the last message-passing layer of GNNs). (3) Our theoretical analysis is applicable to multi-class tasks. Specifically, we identify the two classes with the largest difference in inter-intra vulnerability among all classes. Subsequently, we employ our theoretical analysis to elucidate the reason behind the existence of this largest uneven vulnerability in the target GNNs. (4) Our paper focuses on the issue of link stealing attacks in the context of homophily graphs and GNNs. In the case of heterophily graphs (Zheng et al., 2021), the effectiveness of link stealing attacks may be restricted due to the heterophily property being in conflict with the intuition behind link stealing attacks (i.e., homophily). Nevertheless, note that this limitation is attributable to the current design of link stealing attacks, rather than to our vulnerability analysis.

4.4. Group-based Link Stealing Attacks

In this section, we seek to answer **(RQ 2)**, i.e., "*Could intelligent attackers who are aware of this uneven vulnerability benefit from it?*". Intuitively, given the uneven vulnerability across groups, attackers can devise more effective and customised privacy attacks since the strategy in current link stealing attacks (i.e., using a shred single threshold for all groups) is not optimal. To this end, we propose a paradigm that underpins the group-based link stealing attacks and then present a method that instantiates this attack paradigm.

4.4.1. Group-based Attack Paradigm

Considering that attackers can only query target GNNs in a black-box setting, an intelligent attacker $\mathcal{A}$ can involve querying results (i.e., $f(v_i)$ and $f(v_j)$) in grouping and use different thresholds for different groups to trigger privacy attacks. Specifically, the group-based attack paradigm (GAP) can be expressed as

$$\mathbf{A}_{i,j}^{pred} = \begin{cases} 1 & \text{if } s(i,j) \geq \tau_{g(i,j)}, \\ 0 & \text{if } s(i,j) < \tau_{g(i,j)}, \end{cases} \tag{6}$$

where $\tau_{g(i,j)} \in [0,1]$ indicates the group-based threshold, $g(i,j)$ denotes the grouping result derived from GNN predictions $f(v_i)$ and $f(v_j)$.

4.4.2. An Instantiation of GAP

This section presents our group-based attack method, which instantiates the GAP in Eq. (6). We first present the attacker's knowledge under the black-box setting and then show how to use them in setting different thresholds for different groups.

Knowledge of Attackers. Besides GNN predictions, the knowledge of an intelligent attacker includes *grouping information* and *size difference between groups*. **(1)** Unlike directly using GNN predictions in generic link stealing attacks (i.e., Attack-0 in (He et al., 2021a)), intelligent attackers can use them in grouping node pairs to conduct more effective attacks. For example, for a well-trained GNN model engaged in N-class node prediction, intelligent attackers can divide all node pairs into $\frac{N(N+1)}{2}$ groups. However, crafting a specific threshold for each group can be effort-consuming when N is large. Thus, we propose to divide all node pairs into the inter-class group and intra-class groups, which is a basic grouping manner that is not affected by N. **(2)** The group size difference represents that the size of the intra-class group is generally larger than that of the inter-class group, which indicates that intra-class node pairs are the majority of all node pairs. For example, assuming that the victim GNN model is designed for binary node classification, attackers can obtain $\frac{m^2+n^2}{2}$ intra-class node pairs and mn inter-class node pairs, where m and n indicate the node size of class 0 and class 1, respectively. $(m^2+n^2)/2 \geq mn$ supports that the intra-class group is generally the majority of all node pairs.

Setting of Group Thresholds. Given the above knowledge of attackers, we propose the following group thresholds to launch group-based link stealing attacks, i.e.,

$$\begin{aligned} \tau_{intra} &= \tau_s, \\ \tau_{inter} &= \alpha\tau_s + (1-\alpha)\beta, \end{aligned} \tag{7}$$

where τ_s represents the single threshold obtained from the generic link stealing attacks (i.e., Attack-0 in (He et al., 2021a)). α is a parameter derived from the group sizes and we set $\alpha = \frac{|g_{inter}|}{|g_{intra} \cup g_{inter}|}$. Another parameter β takes into account the homophily of graph data, and we assume that inter-class node pairs are unconnected (i.e., $\beta = 1$) to alleviate the minor role of g_{inter} when calculating τ_s. The intuition of Eq. (7) can be found in our discussion in Appendix A. Note that in this paper, like vanilla link stealing attacks, we assume that target GNNs are trained on homophily graphs and use this common assumption (i.e., homophily of graphs) as prior knowledge to set β.

5. Experiments

In this section, we provide empirical evidence on the existence of uneven vulnerability in current link stealing attacks, followed by evaluating our group-based attack methods.

5.1. Experimental Setup

Datasets. Our evaluations employ ten real-world datasets: Cora (Kipf & Welling, 2017), Citeseer (Kipf & Welling, 2017), Pubmed (Kipf & Welling, 2017), COX2 (Sutherland et al., 2003), DHFR (Sutherland et al., 2003), Enzymes (Dobson & Doig, 2003), Proteins_full (Borgwardt et al., 2005), Credit defaulter graph (Yeh & Lien, 2009), German credit graph (Dua et al., 2017) and Ogbn-Arxiv (Hu et al., 2020). Cora, Citeseer, Pubmed, and Ogbn-Arxiv are citation networks, where nodes denote publication and edges represent citations among nodes. COX2, DHFR, Enzymes, and Proteins come from the chemical community, where nodes indicate molecules, and edges represent their interaction relationship (He et al., 2021a). In the Credit and German datasets, nodes indicate individuals, and they are linked based on their similarity (e.g., spending patterns). These datasets are widely utilised to assess edge privacy risks (e.g., COX2, Enzymes, Proteins (He et al., 2021a)) or the fairness (e.g., Credit, German (Agarwal et al., 2021; Dong et al., 2022)) of GNNs for node classification.

Victim Models and Metrics of Attacking. Following previous link stealing attacks (He et al., 2021a), we choose Graph Convolutional Networks (GCNs, (Kipf & Welling, 2017)) as victim models. Furthermore, we also use Graph Attention Networks (GATs (Velickovic et al., 2018)) and GraphSAGE (Hamilton et al., 2017) models to verify the broad existence of uneven vulnerability in different GNN architectures. The GCNs have 2 hidden layers with 16 units and employ ReLU and softmax as activation functions; the GATs have 2 hidden layers (16 units) with 1 head of attentions and use ELU and softmax as activation functions; the GraphSAGE models have 1 hidden layer (16 units, the Relu activation function) and use 1 MLP layer as the classifier. We use the AUC (area under the ROC curve) as an evaluation metric of attacking performance. Given a specified scoring model, the AUC measures the possibility that, when randomly selecting samples, a positive sample will have a higher score than a negative sample (Fawcett, 2006). We also use the attack success rate (i.e., ASR= $\frac{\#\ Successful\ attacks}{\#\ All\ attacks}$) (Hu et al., 2022) of attack methods to compare our method (i.e., group-based attack methods) and the baseline method (i.e., Attack-0 (He et al., 2021a)).

Others. Following previous attacks (He et al., 2021a), we use cosine, euclidean, correlation, chebyshev, braycurtis, canberra, cityblock and sqeuclidean distance to measure the similarity of two nodes' posteriors. Next, the attackers obtain a link prediction score based on the similarity of two

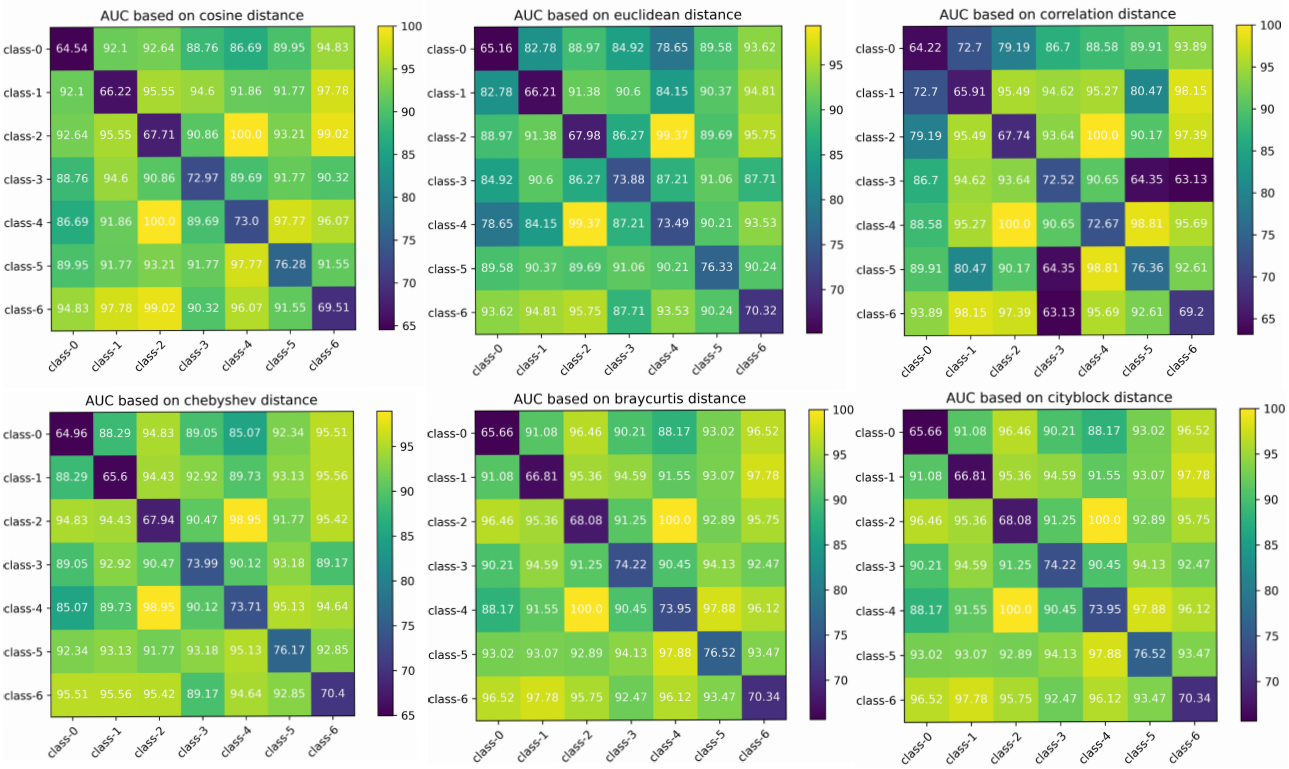

Figure 2. Visualisation of AUC scores across groups in GCNs on the Cora dataset. The x- and y-axis indicate the predicted label of the nodes in a node pair, and the scores in the matrix represent the AUC in link stealing attacks. The difference in AUC between groups demonstrates the existence of uneven vulnerability to link stealing attacks.

nodes. In our group-based attacks, attackers can divide all possible node pairs into intra-class and inter-class groups according to the predicted labels of nodes in node pairs.

5.2. Existence of Uneven Vulnerability

In this section, we conduct generic link stealing attacks (He et al., 2021a) on ten real-world datasets to evaluate whether privacy risks of edges change across different groups, which empirically confirm our theoretical analysis in Section 4.3.

(1) Visualising Privacy Risks of Node Pairs from Different Groups. As shown in Fig. 2, the visualisation of AUC scores demonstrates that node pairs from different groups suffer different degrees of privacy risks. Specifically, we have the following observations. **(a)** In most cases, the AUC scores from inter-class groups are higher than those from intra-class groups. This observation provides empirical evidence for uneven vulnerability across groups (i.e., RQ1), which also confirms our theoretical analysis in Section 4.3. **(b)** In inter-class or intra-class groups, the vulnerability of node pairs with different node labels is also different. For example, in the intra-class groups, the node pairs from (0,0), (1,1), (2,2) and (6,6) groups are less vulnerable than that from (3,3), (4,4), and (5,5) groups across different distances. In all inter-class groups, node pairs from group (2,4) always own the largest vulnerability.

(2) Uneven Vulnerability between g_{inter} and g_{intra}. We evaluate the privacy risk (AUC) difference between groups when dividing all node pairs into intra-class and inter-class groups. **(a)** As shown in Table 2 (Appendix B), the obvious difference in AUC indicates that there is an uneven vulnerability in GCNs. As a follow-up to these evaluations on small datasets, further results on Ogbn-Arxiv (Table 5, Appendix B) indicate that large datasets also exhibit uneven vulnerability. **(b)** Evaluation results on GAT (Table 3, Appendix B) and GraphSAGE (Table 4, Appendix B) confirm the broad existence of uneven vulnerability in different GNN architectures. **(c)** According to Tables 2, 3, and 4, the GraphSAGE models have lower privacy risks and a smaller vulnerability difference between g_{inter} and g_{intra} among the GCN, GAT and GraphSAGE models, potentially resulting from the sampling operation in GraphSAGE alleviates the memorisation of GNN on training graphs.

5.3. Performance of Group-based Attacks

This section illustrates the effectiveness of our group-based link stealing attacks. We first visualise the distribution of

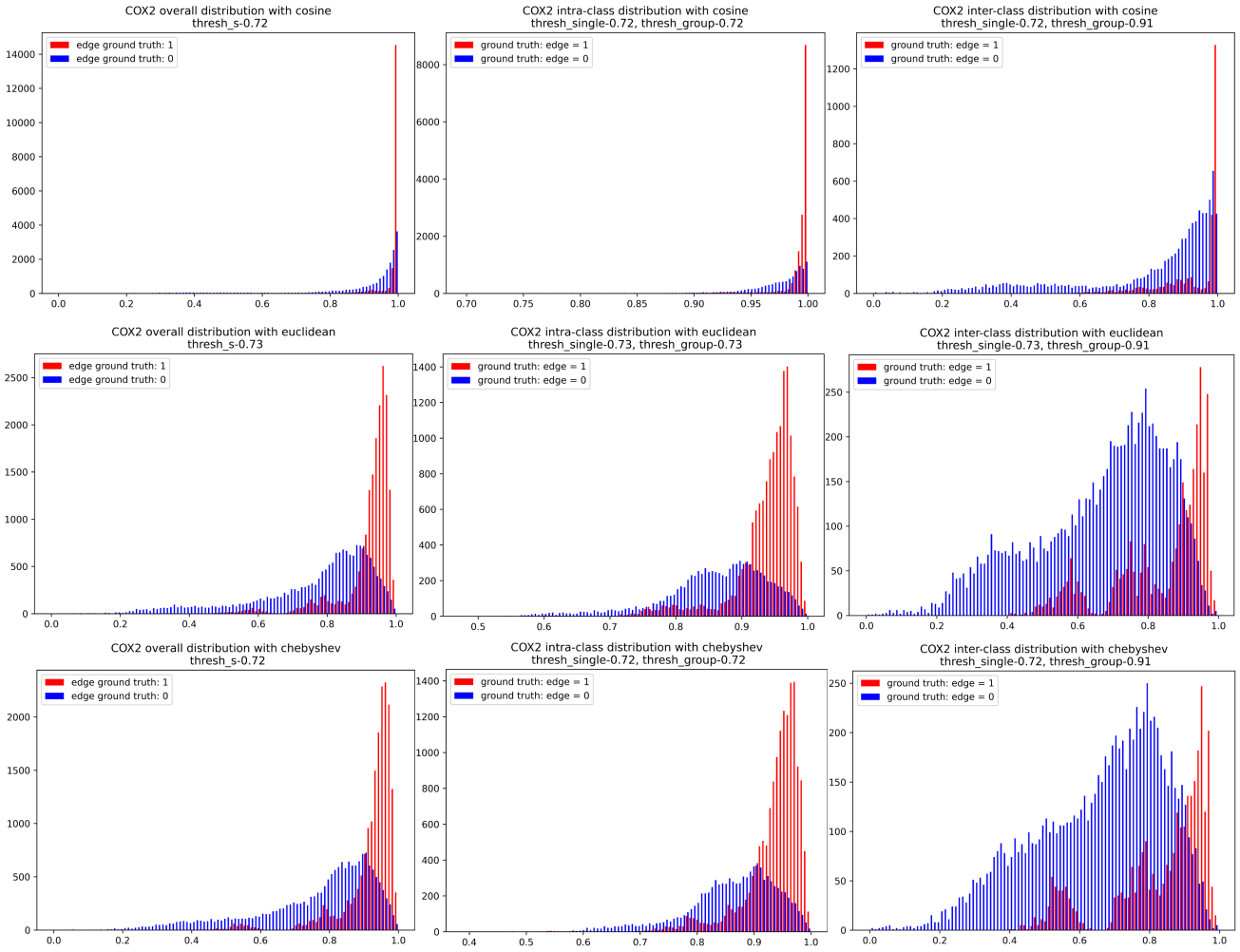

Figure 3. Distribution of prediction score for node pairs in the COX2 dataset. Figures in the left, middle, and right column demonstrate the link prediction score for node pairs in the overall samples, the intra-class group, and the inter-class group, respectively. The first, second and third rows show the score distribution calculated from the cosine, euclidean, and chebyshev distance between two nodes' posteriors from the victim GNN f. In these figures, the x-axis represents the prediction score on node pairs, and the y-axis indicates the number of node pairs with different scores. The red lines indicate that there are edges in the training graph of f, while the blue lines represent the absence of edges.

link prediction score from the view of overall, intra-class, and inter-class node pairs, and then evaluate the attack performance of our method on nine real-world datasets. The main observations are listed as follows.

(1) Distribution Similarity. As shown in Fig. 3, our group-based method has a better threshold setting than generic attack methods based on a single shared threshold. The score distribution derived from intra-class groups is similar to that from overall samples, while the score on inter-class groups has a different distribution. This observation is consistent with our analysis in Section 4.4.2, which shows that the intra-class group is the majority of the overall samples and empirically supports the threshold setting (i.e., $\tau_{intra} = \tau_s$) in Eq. (7). Noting that although the left and middle columns in Fig. 3 show that τ_s may not be optimal, inferring the threshold by K-means is still an effective and practical method for attackers in a black-box setting.

(2) Better Threshold. For the inter-class group, our group-based method obtains a better threshold than τ_s, since it is closer to the optimal decision boundary (i.e., the x-axis value at the interaction position of the red and blue distributions). According to the right column of Fig. 3, the threshold obtained from the overall samples underestimates the privacy risk of node pairs in the inter-class group. For example, the threshold τ_s is 0.73 when using the euclidean distance (see the distribution in the second row and right

Table 1. Comparison of Attack Success Rate (i.e., ASR) between Our Method and Generic Link Stealing Attacks.

Datasets		*S*	*G*	*S*	*G*	*S*	*G*	*S*	*G*	ΔASR ↑
COX2		cosine		euclidean		correlation		chebyshev		18.40
	g_{inter}	41.50	**58.61**	61.01	**83.17**	41.20	**55.37**	63.35	**82.25**	
	overall	55.29	**60.73**	63.13	**70.18**	55.21	**59.72**	63.79	**69.81**	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	58.24	**84.19**	78.22	**81.08**	58.24	**84.19**	43.92	**64.05**	
	overall	62.35	**70.61**	77.96	**78.87**	62.35	**70.61**	56.06	**62.47**	
DHFR		cosine		euclidean		correlation		chebyshev		18.26
	g_{inter}	52.44	**88.48**	76.47	**86.27**	51.70	**87.94**	72.88	**86.70**	
	overall	64.80	**74.19**	77.87	**80.43**	64.31	**73.77**	76.58	**80.19**	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	80.23	**85.76**	80.28	**82.94**	80.23	**85.76**	50.90	**87.39**	
	overall	79.34	**80.78**	83.77	**84.46**	79.34	**80.78**	64.51	**74.03**	
PROTEINS		cosine		euclidean		correlation		chebyshev		29.86
	g_{inter}	63.39	**98.22**	68.89	**97.91**	70.43	**98.14**	68.11	**97.91**	
	overall	51.10	**51.41**	51.19	**51.45**	50.61	**50.85**	51.18	**51.45**	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	68.11	**97.91**	78.56	**97.91**	68.11	**97.91**	59.75	**98.30**	
	overall	51.18	**51.45**	51.32	**51.49**	51.18	**51.45**	51.06	**51.40**	

* In this table, "*S*" and "*G*" indicate the generic link stealing attacks and our group-based method, respectively. "g_{inter}" and "*overall*" represent the inter-class group and overall node pairs, respectively. ΔASR indicates the average improvement in attack performance of our method on g_{inter}. We only show "g_{inter}" results because the "g_{intra}" in our method has the same threshold as vanilla link stealing attacks, resulting in the same ASR when inferring edges.

column in Fig. 3). Using τ_s, attackers predict that there exists an edge when the score of a pair of nodes is higher than τ_s, even if it is more likely to come from the node pairs with no edges. However, with setting $\tau_{inter} = 0.91$, our method can produce fewer classification errors from the view of statistical decision (Duda et al., 2001).

(3) Higher Attack Benefits. In addition to visualising the obtained thresholds, we compare our group-based attack method with generic link stealing attacks on nine real-world datasets to evaluate its practical attack performance. As shown in Tables 1 and 6, our group-based attack method consistently outperforms generic link stealing attacks (He et al., 2021a) on most datasets. For example, when using cosine distance, the inter-class prediction accuracy on the Proteins dataset increases by 34.83 (from 63.39 to 98.22) when employing τ_{inter}, which reveals the effectiveness of our group-based attack method and the serious privacy risks of node pairs in g_{inter}. It is worth noting that current evaluations of our method, whose grouping operation depends on node predictions, are derived from GNNs that are not 100% accurate, indicating that our method can potentially be improved if the target GNNs are more accurate.

6. Conclusion

In this paper, we investigate the uneven vulnerability across different groups to link stealing attacks on GNNs. We first illustrate theoretical evidence on the vulnerability difference between intra-class and inter-class groups, which inspires us to devise group-based methods to achieve more effective link stealing attacks. Our experimental evaluations validate the uneven vulnerability across groups and show the superior performance of the proposed group-based methods.

Our future work mainly includes exploring other types of uneven vulnerability, which helps GNN researchers and practitioners understand the practical privacy risks of GNNs. Furthermore, current fairness studies on GNNs focus on ensuring fair model performance (e.g., accuracy) on similar individuals or vulnerable groups (e.g., female users). Designing methods to mitigate this unfair vulnerability is another promising direction for building a fair GNN with respect to privacy risks.

Acknowledgements

This research was supported by an Australian Research Council (ARC) Future Fellowship (FT210100097), and a Monash-Data61 collaborative research project on Novel security solutions in the context of AI and potential quantum supremacy.

References

Adeshina, S. Detecting fraud in heterogeneous networks using amazon sagemaker and deep graph library, Jun 2020. URL http://surl.li/coqjv.

Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation learning. In *UAI*, volume 161 of *Proceedings of Machine Learning Research*, pp. 2114–2124. AUAI Press, 2021.

Backstrom, L. and Leskovec, J. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pp. 635–644. ACM, 2011.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR (Poster)*. OpenReview.net, 2018.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S. V. N., Smola, A. J., and Kriegel, H. Protein function prediction via graph kernels. In *ISMB (Supplement of Bioinformatics)*, pp. 47–56, 2005.

Chang, H. and Shokri, R. On the privacy risks of algorithmic fairness. In *EuroS&P*, pp. 292–303. IEEE, 2021.

Chen, Z., Li, P., Liu, H., and Hong, P. Characterizing the influence of graph elements. In *ICLR (Poster)*. OpenReview.net, 2023.

Cherubin, G., Chatzikokolakis, K., and Palamidessi, C. F-BLEAU: fast black-box leakage estimation. In *IEEE S&P*, pp. 835–852. IEEE, 2019.

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

Dong, Y., Ma, J., Chen, C., and Li, J. Fairness in graph mining: A survey. *CoRR*, abs/2204.09888, 2022.

Dua, D., Graff, C., et al. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern classification, 2nd Edition*. Wiley, 2001.

Duddu, V., Boutet, A., and Shejwalkar, V. Quantifying privacy leakage in graph embedding. In *MobiQuitous*, pp. 76–85. ACM, 2020.

Egilmez, H. E., Pavez, E., and Ortega, A. Graph learning from data under laplacian and structural constraints. *IEEE J. Sel. Top. Signal Process.*, 11(6):825–841, 2017.

Farokhi, F. and Kâafar, M. A. Modelling and quantifying membership information leakage in machine learning. *CoRR*, abs/2001.10648, 2020.

Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.*, 27(8):861–874, 2006.

Galmés, M. F., Rusek, K., Suárez-Varela, J., Xiao, S., Shi, X., Cheng, X., Wu, B., Barlet-Ros, P., and Cabellos-Aparicio, A. Routenet-erlang: A graph neural network for network performance evaluation. In *INFOCOM*, pp. 2018–2027. IEEE, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *NeurIPS*, pp. 1024–1034, 2017.

He, S., Liu, K., Ji, G., and Zhao, J. Learning to represent knowledge graphs with gaussian embedding. In *CIKM*, pp. 623–632. ACM, 2015.

He, X., Jia, J., Backes, M., Gong, N. Z., and Zhang, Y. Stealing links from graph neural networks. In *USENIX Security Symposium*, pp. 2669–2686. USENIX Association, 2021a.

He, X., Wen, R., Wu, Y., Backes, M., Shen, Y., and Zhang, Y. Node-level membership inference attacks against graph neural networks. *CoRR*, abs/2102.05429, 2021b.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., and Zhang, X. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.*, 54(11s):235:1–235:37, 2022.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Humphries, T., Rafuse, M., Tulloch, L., Oya, S., Goldberg, I., and Kerschbaum, F. Differentially private learning does not bound membership inference. *CoRR*, abs/2010.12112, 2020.

Jayaraman, B., Wang, L., Knipmeyer, K., Gu, Q., and Evans, D. Revisiting membership inference under realistic assumptions. *Proc. Priv. Enhancing Technol.*, 2021(2): 348–368, 2021.

Jin, M., Li, Y., and Pan, S. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. In *NeurIPS*, 2022a.

Jin, M., Zheng, Y., Li, Y., Chen, S., Yang, B., and Pan, S. Multivariate time series forecasting with dynamic graph neural odes. *IEEE TKDE*, 2022b.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net, 2017.

Kulynych, B., Yaghini, M., Cherubin, G., Veale, M., and Troncoso, C. Disparate vulnerability to membership inference attacks. *Proc. Priv. Enhancing Technol.*, 2022(1): 460–480, 2022.

Lackey, B. Analyze graph data on google cloud with neo4j and vertex ai, 01 2022. URL http://surl.li/coqkg.

Li, P., Wang, Y., Zhao, H., Hong, P., and Liu, H. On dyadic fairness: Exploring and mitigating bias in graph connections. In *ICLR*. OpenReview.net, 2021.

Li, X. and Liu, B. Maximum entropy principle for fuzzy variables. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 15(Supplement-2):43–52, 2007.

Liu, Y., Ding, K., Liu, H., and Pan, S. GOOD-D: on unsupervised graph out-of-distribution detection. In *WSDM*, pp. 339–347. ACM, 2023a.

Liu, Y., Ding, K., Wang, J., Lee, V., Liu, H., and Pan, S. Learning strong graph neural networks with weak information. In *KDD*. ACM, 2023b.

Liu, Y., Zheng, Y., Zhang, D., Lee, V. C. S., and Pan, S. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *AAAI*, 2023c.

Luo, Z., Bao, Y., and Wu, C. Optimizing task placement and online scheduling for distributed GNN training acceleration. In *INFOCOM*, pp. 890–899. IEEE, 2022.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2021.

Mondal, S., Basu, A., and Mukherjee, N. Building a trust-based doctor recommendation system on top of multilayer graph database. *J. Biomed. Informatics*, 110:103549, 2020.

Murakonda, S. K., Shokri, R., and Theodorakopoulos, G. Quantifying the privacy risks of learning high-dimensional graphical models. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2287–2295. PMLR, 2021.

Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Pal, A., Eksombatchai, C., Zhou, Y., Zhao, B., Rosenberg, C., and Leskovec, J. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In *KDD*, pp. 2311–2320. ACM, 2020.

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., and Zhang, C. Adversarially regularized graph autoencoder for graph embedding. In *IJCAI*, pp. 2609–2615. ijcai.org, 2018.

Pan, S., Hu, R., Fung, S., Long, G., Jiang, J., and Zhang, C. Learning graph embedding with adversarial training methods. *IEEE Trans. Cybern.*, 50(6):2475–2487, 2020.

Saxena, N. A., Huang, K., DeFilippis, E., Radanovic, G., Parkes, D. C., and Liu, Y. How do fairness definitions fare?: Examining public attitudes towards algorithmic definitions of fairness. In *AIES*, pp. 99–106. ACM, 2019.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *IEEE S&P*, pp. 3–18. IEEE Computer Society, 2017.

Sutherland, J. J., O'Brien, L. A., and Weaver, D. F. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *J. Chem. Inf. Comput. Sci.*, 43(6):1906–1915, 2003.

Tan, Y., Liu, Y., Long, G., Jiang, J., Lu, Q., and Zhang, C. Federated learning on non-iid graphs via structural knowledge sharing. In *AAAI*, 2023.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *ICLR (Poster)*. OpenReview.net, 2018.

Wan, S., Zhan, Y., Liu, L., Yu, B., Pan, S., and Gong, C. Contrastive graph poisson networks: Semi-supervised learning with extremely limited labels. In *NeurIPS*, pp. 6316–6327, 2021.

Wang, M., Hui, L., Cui, Y., Liang, R., and Li, Z. xnet: Improving expressiveness and granularity for network modeling with graph neural networks. In *INFOCOM*, pp. 2028–2037. IEEE, 2022.

Wang, Y. and Sun, L. Membership inference attacks on knowledge graphs. *CoRR*, abs/2104.08273, 2021.

Wu, B., Yang, X., Pan, S., and Yuan, X. Adapting membership inference attacks to GNN for graph classification: Approaches and implications. In *ICDM*. IEEE Computer Society, 2021a.

Wu, B., Yang, X., Pan, S., and Yuan, X. Model extraction attacks on graph neural networks: Taxonomy and realization. In *AsiaCCS*. ACM, 2022a.

Wu, F., Long, Y., Zhang, C., and Li, B. LINKTELLER: recovering private edges from graph neural networks via influence analysis. In *IEEE Symposium on Security and Privacy*, pp. 2005–2024. IEEE, 2022b.

Wu, Y., Wu, Z., Wu, Q., Ge, Z., and Cai, J. Exploring smoothness and class-separation for semi-supervised medical image segmentation. In *MICCAI (5)*, volume 13435 of *Lecture Notes in Computer Science*, pp. 34–43. Springer, 2022c.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE TNNLS*, 32(1):4–24, 2021b.

Yeh, I. and Lien, C. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst. Appl.*, 36(2): 2473–2480, 2009.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *CSF*, pp. 268–282. IEEE Computer Society, 2018.

Zhang, H., Wu, B., Yang, X., Zhou, C., Wang, S., Yuan, X., and Pan, S. Projective ranking: A transferable evasion attack method on graph neural networks. In *CIKM*, pp. 3617–3621. ACM, 2021.

Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., and Pei, J. Trustworthy graph neural networks: Aspects, methods and trends. *CoRR*, abs/2205.07424, 2022a.

Zhang, H., Yuan, X., Zhou, C., and Pan, S. Projective ranking-based gnn evasion attacks. *IEEE TKDE*, 2022b.

Zhang, H., Yuan, X., Nguyen, Q. V. H., and Pan, S. On the interaction between node fairness and edge privacy in graph neural networks. *CoRR*, abs/2301.12951, 2023.

Zheng, X., Liu, Y., Pan, S., Zhang, M., Jin, D., and Yu, P. S. Graph neural networks for graphs with heterophily: A survey. *CoRR*, abs/2202.07082, 2022a.

Zheng, X., Zhang, M., Chen, C., Li, C., Zhou, C., and Pan, S. Multi-relational graph neural architecture search with fine-grained message passing. In *ICDM*, pp. 783–792. IEEE, 2022b.

Zheng, X., Zhang, M., Chen, C., Zhang, Q., Zhou, C., and Pan, S. Auto-heg: Automated graph neural network on heterophilic graphs. In *WWW*, pp. 611–620. ACM, 2023.

Zheng, Y., Lee, V. C. S., Wu, Z., and Pan, S. Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction. In *PAKDD (1)*, volume 12712 of *Lecture Notes in Computer Science*, pp. 140–151. Springer, 2021.

Zheng, Y., Pan, S., Lee, V. C. S., Zheng, Y., and Yu, P. S. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. In *NeurIPS*, 2022c.

Zheng, Y., Zheng, Y., Zhou, X., Gong, C., Lee, V. C. S., and Pan, S. Unifying graph contrastive learning with flexible contextual scopes. In *ICDM*, pp. 793–802. IEEE, 2022d.

Zhong, D., Sun, H., Xu, J., Gong, N. Z., and Wang, W. H. Understanding disparate effects of membership inference attacks and their countermeasures. In *AsiaCCS*, pp. 959–974. ACM, 2022.

Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed, N. K., and Koutra, D. Graph neural networks with heterophily. In *AAAI*, pp. 11168–11176. AAAI Press, 2021.

A. Proof of Theorem 4.1.

To demonstrate the vulnerability difference between node pairs from $g_0 = \{\mathbf{A}_{i,j} \mid S(v_i) \neq S(v_j)\}$ and $g_1 = \{\mathbf{A}_{i,j} \mid S(v_i) = S(v_j)\}$, we first derive an equivalent form of vulnerability metric $V(f)$, which is used in our following proof. In the the likelihood-ratio test, the relationship between α and β can be expressed as

$$\mu_1^e - z_\beta \sigma^e = \mu_0^e + z_{1-\alpha}\sigma^e, \tag{8}$$

which can be derived by $z_{1-\alpha}$ and z_β own a shared interaction position of $N(\mu_0^e, \sigma^e)$ and $N(\mu_1^e, \sigma^e)$. So we have

$$V(f) = z_{1-\alpha} + z_\beta = \frac{\mu_1^e - \mu_0^e}{\sigma^e}. \tag{9}$$

Next, we will prove $\Delta V_{g_0,g_1}(f) = \mid V_{g_0}(f) - V_{g_1}(f) \mid \neq 0$. As shown in follows, according to Eq. (9), the uneven vulnerability can be derived by calculating the distance sensitivity difference between node pairs from g_0 and g_1 due to the existence of an edge.

(1) Case $g_1 = \{\mathbf{A}_{i,j} \mid S(v_i) = S(v_j)\}$: As an instance of the message passing in Eq. (1), the one-hop mean-aggregation operation can be expressed as $\widehat{\mathbf{A}}\mathbf{X}$. From the view of an individual node v_i, the message passing operation is

$$\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i = \frac{1}{d_i+1}\left(\mathbf{X}[t]_i + \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{X}[t]_j\right) = \frac{1}{d_i+1}\left(\mathbf{X}[t]_i + \sum_{\substack{v_j \in \mathcal{N}(v_i) \\ m_j=0}} \mathbf{X}[t]_j^{y_0} + \sum_{\substack{v_j \in \mathcal{N}(v_i) \\ m_j=1}} \mathbf{X}[t]_j^{y_1}\right). \tag{10}$$

Let $d_i^{y_0}$ and $d_i^{y_1}$ denote the number of nodes from $\mathbf{X}^{y_0}$ and $\mathbf{X}^{y_1}$, we have $d_i = d_i^{y_0} + d_i^{y_1}$ and $\sum_{\substack{v_j \in \mathcal{N}(v_i) \\ m_j=0}} \mathbf{X}[t]_j^{y_0} \sim N(d_i^{y_0}\mu_0^e, d_i^{y_0}\sigma^e)$, $\sum_{\substack{v_j \in \mathcal{N}(v_i) \\ m_j=1}} \mathbf{X}[t]_j^{y_0} \sim N(d_i^{y_1}\mu_1^e, d_i^{y_1}\sigma^e)$. Without loss of generality, here we assume that node pair $(v_i, v_j) \in g_1^0 = \{\mathbf{A}_{i,j} = 0 \mid S(v_i) = S(v_j)\}$ and $S(v_i) = 0$, then $\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i$ and $\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j$ can be approximately expressed as

$$\begin{aligned}
\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^0} &\approx \frac{1}{d_i+1}\left(\mathbf{X}[t]_i + d_i^{y_0}\mu_0^e + d_i^{y_1}\mu_1^e\right), \\
\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^0} &\approx \frac{1}{d_j+1}\left(\mathbf{X}[t]_j + d_j^{y_0}\mu_0^e + d_j^{y_1}\mu_1^e\right).
\end{aligned} \tag{11}$$

When adding an edge between v_i and v_j, node pair $(v_i, v_j) \in g_1^1 = \{\mathbf{A}_{i,j} = 1 \mid S(v_i) = S(v_j)\}$, the embedding $\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i$ and $\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j$ can be approximately expressed as

$$\begin{aligned}
\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^1} &\approx \frac{1}{d_i+2}\left(\mathbf{X}[t]_i + \mathbf{X}[t]_j + d_i^{y_0}\mu_0^e + d_i^{y_1}\mu_1^e\right), \\
\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^1} &\approx \frac{1}{d_j+2}\left(\mathbf{X}[t]_j + \mathbf{X}[t]_i + d_j^{y_0}\mu_0^e + d_j^{y_1}\mu_1^e\right).
\end{aligned} \tag{12}$$

Assuming that the distances between v_i and v_j are expressed as

$$\begin{aligned}
d_{g_1^1}(v_i, v_j) &= \left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^1} - \left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^1}, \\
d_{g_1^0}(v_i, v_j) &= \left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^0} - \left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^0},
\end{aligned} \tag{13}$$

the distance sensitivity $\Delta d(v_i, v_j)$ on group g_1 with respect to the existence of edge e_{ij} can be calculated as

$$\Delta d_{g_1}(v_i, v_j) = \left| d_{g_1^0}(v_i, v_j) - d_{g_1^1}(v_i, v_j) \right| = \left| \frac{1}{d_i+2}\left(\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^0} - \mathbf{X}[t]_j\right) - \frac{1}{d_j+2}\left(\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^0} - \mathbf{X}[t]_i\right) \right|. \tag{14}$$

Since

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_1^0} - \mathbf{X}[t]_j\right] &= \frac{(1+d_i^{y_0})\mu_0^e + d_i^{y_1}\mu_1^e}{d_i+1} - \mu_0^e = \frac{d_i^{y_1}(\mu_1^e-\mu_0^e)}{d_i+1},\\ \mathbb{E}\left[\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_1^0} - \mathbf{X}[t]_i\right] &= \frac{(1+d_j^{y_0})\mu_0^e + d_j^{y_1}\mu_1^e}{d_j+1} - \mu_0^e = \frac{d_j^{y_1}(\mu_1^e-\mu_0^e)}{d_j+1},\end{aligned} \tag{15}$$

we have $\mathbb{E}\left[\Delta d_{g_1}(v_i,v_j)\right] = |(\mu_1^e-\mu_0^e)\delta_{g_1}|$, where $\delta_{g_1} = \frac{d_i^{y_1}}{(d_i+1)(d_i+2)} - \frac{d_j^{y_1}}{(d_j+1)(d_j+2)}$.

(2) Case $g_0 = \{\mathbf{A}_{i,j} \mid S(v_i) \neq S(v_j)\}$. Following a similar way, we obtain the expectation of $\Delta d_{g_0}(v_i,v_j)$. Without loss of generality, assuming that $S(v_i)=0$ and $S(v_j)=1$, we have

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_i^{g_0^0} - \mathbf{X}[t]_j\right] &= \frac{(1+d_i^{y_0})\mu_0^e + d_i^{y_1}\mu_1^e}{d_i+1} - \mu_1^e = \frac{(1+d_i^{y_0})(\mu_0^e-\mu_1^e)}{d_i+1},\\ \mathbb{E}\left[\left(\widehat{\mathbf{A}}\mathbf{X}[t]\right)_j^{g_0^0} - \mathbf{X}[t]_i\right] &= \frac{(1+d_j^{y_1})\mu_1^e + d_j^{y_0}\mu_0^e}{d_j+1} - \mu_0^e = \frac{(1+d_j^{y_1})(\mu_1^e-\mu_0^e)}{d_j+1},\end{aligned} \tag{16}$$

and $\mathbb{E}\left[\Delta d_{g_0}(v_i,v_j)\right] = |(\mu_1^e-\mu_0^e)\delta_{g_0}|$, where $\delta_{g_0} = \frac{1+d_i^{y_0}}{(d_i+1)(d_i+2)} + \frac{1+d_j^{y_1}}{(d_j+1)(d_j+2)}$.

Given $\mathbb{E}\left[\Delta d_{g_0}(v_i,v_j)\right]$ and $\mathbb{E}\left[\Delta d_{g_1}(v_i,v_j)\right]$, we have

$$\mathbb{E}\left[\Delta V_{g_0,g_1}(f)\right] = \mathbb{E}\left[|V_{g_0}(f) - V_{g_1}(f)|\right] = \frac{\left|\mathbb{E}\left[\Delta d_{g_0}(v_i,v_j)\right] - \mathbb{E}\left[\Delta d_{g_1}(v_i,v_j)\right]\right|}{\sigma^e} = \frac{|(\mu_1^e-\mu_0^e)|}{\sigma^e}\left|\mathbb{E}\left[\delta_{g_0}\right] - \mathbb{E}\left[\delta_{g_1}\right]\right|. \tag{17}$$

For a well-trained GNN f, its competent performance in discriminating node labels indicates that $|(\mu_1^e-\mu_0^e)| \neq 0$. Moreover, according to current literature on graph data (Zhu et al., 2021; Zheng et al., 2022a; 2023) and link stealing attacks (He et al., 2021a), the homophily of graph data implies nodes from the same class are more likely to connect each other (i.e., generally $\frac{1+d_i^{y_0}}{(d_i+1)}$ in δ_{g_0} is larger than $\frac{d_i^{y_1}}{(d_i+1)}$ in δ_{g_1}, and $\frac{1+d_j^{y_1}}{(d_j+1)}$ in δ_{g_0} is larger than $\frac{d_j^{y_1}}{(d_j+1)}$ in δ_{g_1}), which indicates $|\mathbb{E}\left[\delta_{g_0}\right] - \mathbb{E}\left[\delta_{g_1}\right]| \neq 0$. Therefore, we obtain that $\mathbb{E}\left[\Delta V_{g_0,g_1}(f)\right] \neq 0$.

A.1. Discussions

Based on our above proof, some insights into understanding the edge privacy risk can be derived from Eq. (17).

(1) Trade-off between GNN performance and edge privacy. Our analysis results in cases g_0 and g_1 show that model performance potentially contributes to the edge privacy risk, since the item $|\mu_1^e-\mu_0^e|$ indicates that the better a model's discrimination ability with respect to node classification, the higher its privacy risk of edges in the training graph. This insight is consistent with the homophily property (i.e., similar nodes are likely to connect to each other) of most graph data, which leads to that edges can be inferred once the node labels are exactly predicted by GNNs with competent performance. In addition to potentially explaining why competent GNNs are vulnerable to link stealing attacks, our analysis also reveals that there exists a trade-off between GNN performance and edge privacy.

(2) Underestimated privacy risk of g_0. As we demonstrated in Section 4.4.2 (i.e., the knowledge of attackers), the intra-class group dominates node pairs that are used in calculating the shared single threshold, which results in the obtained threshold is more suitable for attacking the intra-class group and not optimal for the inter-class group. Considering the homophily property of graph data and connected nodes are prone to have closer embedding and predictions, node pairs in the inter-class group (i.e., g_0) generally have a larger distance than that from the intra-class group (i.e., g_1). Thus, improving the threshold value helps reduce the error of classifying node pairs from g_0 as that from g_1. Moreover, when setting the threshold for the inter-class group, the size ratio $\frac{|g_{inter}|}{|g_{intra}\cup g_{inter}|}$ should be taken into consideration since it potentially reflects the bias degree to g_1 when calculating the shared single threshold. Given the above considerations, we propose the parameter setting of our group-based attacks in Section 4.4.2.

B. Additional Experimental Results.

This section presents supplementary experimental results for Section 5. In this paper, we adopt AUC and ASR as metrics to assess the privacy risks of edges in GNNs. It should be noted that AUC scores quantify the privacy risk of edges when attackers attempt to infer the structure of the training graph from the prediction similarity of node pairs. ASR evaluates the practical attack benefits (i.e., number of successful attacks on both connected and unconnected node pairs) when using different attack methods. From the perspective of privacy and security, both connected and unconnected status in node pairs are confidential for target GNNs and should be accorded equal protection. Although there may be a quantity disparity between connected and unconnected node pairs in practical graphs, ASR provides meaningful and intuitive evaluations of the attack benefits of different methods.

Table 2. Comparison of Privacy Risks (AUC) between Intra-class and Inter-class Node Pairs on GCN models.

Datasets		Cosi	Eucl	Corr	Cheb	Bray	Canb	City	Sqeu
Cora	Intra-class	69.74	70.39	69.48	70.38	70.59	82.29	70.59	70.39
	Inter-class	92.62	88.98	90.93	92.22	93.46	84.05	93.46	88.98
Citeseer	Intra-class	73.79	74.68	73.3	74.85	75.07	87.51	75.07	74.68
	Inter-class	93.37	89.86	93.83	93.04	94.06	90.85	94.06	89.86
Pubmed	Intra-class	77.68	79.14	74.29	79.40	79.40	85.80	79.40	79.14
	Inter-class	93.19	93.58	83.48	93.97	93.97	94.69	93.97	93.58
COX2	Intra-class	82.6	82.77	82.18	81.54	84.13	86.89	84.13	82.77
	Inter-class	76.5	79.61	74.73	80.02	80.17	92.77	80.17	79.61
DHFR	Intra-class	93.06	93.63	92.60	92.84	94.26	92.54	94.26	93.63
	Inter-class	97.49	97.57	97.36	96.88	97.93	92.37	97.93	97.57
Enzymes	Intra-class	75.20	75.39	74.48	75.56	75.56	84.94	75.56	75.39
	Inter-class	86.13	86.16	85.97	86.22	86.22	88.91	86.22	86.16
Proteins	Intra-class	50.08	50.06	50.08	50.06	50.06	50.06	50.06	50.06
	Inter-class	96.50	96.10	95.88	96.26	96.26	93.05	96.26	96.10
Credit	Intra-class	77.60	77.55	50.04	77.55	77.55	77.39	77.55	77.55
	Inter-class	53.20	51.99	54.15	51.99	51.99	52.27	51.99	51.99
German	Intra-class	62.80	63.42	50.20	63.42	63.42	63.76	63.42	63.42
	Inter-class	81.85	82.23	50.02	82.23	82.23	82.49	82.23	82.23

* In this table, Cosi, Eucl, Corr, Cheb, Bray, Canb, City, and Sqeu indicate the cosine, euclidean, correlation, chebyshev, braycurtis, canberra, cityblock, sqeuclidean distance respectively when launching generic link stealing attacks (He et al., 2021a).

Table 3. Comparison of Privacy Risks (AUC) between Intra-class and Inter-class Node Pairs on GAT models.

Dataset		Cosi	Eucl	Corr	Cheb	Bray	Canb	City	Sqeu
Cora	Intra-class	68.99	70.56	67.04	70.81	70.89	80.48	70.89	70.56
	Inter-class	92.84	90.46	91.13	92.73	93.76	82.98	93.76	90.46
Citeseer	Intra-class	76.59	77.54	75.83	77.71	77.87	86.99	77.87	77.54
	Inter-class	92.64	89.37	92.75	92.66	93.58	92.22	93.58	89.37
Pubmed	Intra-class	78.97	80.22	74.33	80.46	80.46	84.11	80.46	80.22
	Inter-class	93.60	93.78	84.24	94.13	94.13	94.69	94.13	93.78

Table 4. Comparison of Privacy Risks (AUC) between Intra-class and Inter-class Node Pairs on GraphSAGE models.

Dataset		Cosi	Eucl	Corr	Cheb	Bray	Canb	City	Sqeu
Cora	Intra-class	61.23	54.16	61.09	54.28	54.87	57.15	53.84	54.16
	Inter-class	69.55	62.38	66.93	62.04	67.23	64.26	63.10	62.38
Citeseer	Intra-class	65.08	58.75	65.26	58.85	51.51	54.74	58.40	58.75
	Inter-class	68.87	57.92	67.49	57.60	66.09	64.57	58.73	57.92
Pubmed	Intra-class	65.72	54.92	64.83	54.19	61.15	64.95	56.53	54.92
	Inter-class	80.60	77.85	71.02	76.62	80.46	80.50	79.08	77.85

Table 5. Comparison of Privacy Risks (AUC) between Intra-class and Inter-class Node Pairs on GCN model with the large-scale dataset.

Dataset		Cosi	Eucl	Corr	Cheb	Bray	Canb	City	Sqeu
Ogbn-Arxiv	Intra-class	90.50	76.23	92.37	72.62	84.12	51.77	84.12	76.23
	Inter-class	82.07	72.27	83.40	71.74	79.05	55.66	79.05	72.27

Table 6. Comparison of Attack Success Rate (i.e., ASR) between Our Method and Generic Link Stealing Attacks.

Datasets		*S*	*G*	*S*	*G*	*S*	*G*	*S*	*G*	ΔASR ↑
ENZYMES		cosine		euclidean		correlation		chebyshev		7.21
	g_{inter}	65.82	**83.63**	79.61	**81.92**	71.92	**83.77**	79.78	**81.94**	
	overall	64.08	**69.53**	70.33	**71.04**	65.88	**69.50**	70.41	**71.07**	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	79.78	**81.94**	**85.22**	81.64	79.78	**81.94**	59.81	**82.59**	
	overall	70.41	**71.07**	**81.00**	79.90	70.41	**71.07**	62.36	**69.33**	
Pubmed		cosine		euclidean		correlation		chebyshev		1.71
	g_{inter}	86.31	**92.48**	**92.06**	90.82	88.79	**91.37**	**92.29**	90.78	
	overall	77.26	**79.48**	**80.17**	79.73	78.24	**79.17**	**80.33**	79.78	
		braycurtis		canberra		cityblock		sqeuclidean		
	Inter-class	**92.29**	90.78	**92.80**	90.85	**92.29**	90.78	80.21	**92.84**	
	overall	**80.33**	79.78	**84.78**	84.08	**80.33**	79.78	75.06	**79.61**	
Cora		cosine		euclidean		correlation		chebyshev		-1.93
	g_{inter}	**89.95**	88.20	**89.49**	87.18	**89.47**	87.74	**89.34**	87.11	
	overall	**86.71**	85.86	**86.68**	85.57	**86.47**	85.64	**86.61**	85.54	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	**89.62**	86.98	**90.10**	87.33	**89.62**	86.98	88.84	**89.49**	
	overall	**86.79**	85.52	**83.43**	82.09	**86.79**	85.52	86.17	**86.48**	
Citeseer		cosine		euclidean		correlation		chebyshev		-1.17
	g_{inter}	**92.23**	91.32	**92.42**	90.77	**92.12**	91.05	**92.59**	90.74	
	overall	**86.54**	86.13	**87.04**	86.30	**86.49**	86.01	**87.30**	86.49	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	**92.73**	90.74	**94.16**	91.18	**92.73**	90.74	89.15	**92.20**	
	overall	**87.38**	86.50	**87.73**	86.41	**87.38**	86.50	85.17	**86.52**	
German		cosine		euclidean		correlation		chebyshev		0.15
	g_{inter}	72.02	**77.69**	**78.20**	74.93	74.31	74.31	**78.20**	74.93	
	overall	63.71	**65.38**	**67.33**	66.70	64.30	64.30	**67.33**	66.37	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	**78.20**	74.93	**78.17**	74.93	**78.20**	74.93	66.54	**78.40**	
	overall	**67.33**	66.37	**68.58**	67.62	**67.33**	66.37	62.12	**65.61**	
Credit		cosine		euclidean		correlation		chebyshev		8.45
	g_{inter}	66.19	**92.09**	**92.81**	92.09	92.09	92.09	**92.81**	92.09	
	overall	50.05	**50.07**	**50.91**	50.91	50.07	50.07	**50.91**	50.91	
		braycurtis		canberra		cityblock		sqeuclidean		
	g_{inter}	**92.81**	92.09	73.38	**92.09**	**92.81**	92.09	66.19	**92.09**	
	overall	**50.91**	50.91	50.62	**50.63**	**50.91**	50.91	50.05	**50.07**	

* In this table, "*S*" and "*G*" indicate the generic link stealing attacks and our group-based method, respectively. "g_{inter}" and "*overall*" represent the inter-class group and overall node pairs, respectively. ΔASR indicates the average attack performance improvement of our group-based method on g_{inter}.