Learning Privacy-Preserving Graph Convolutional Network with Partially Observed Sensitive Attributes

Hui Hu  
Department of Computer Science  
University of Wyoming  
hhu1@uwyo.edu

Jayden Parker Vap  
Department of Electrical Engineering  
University of Wyoming  
jvap2@uwyo.edu

Lu Cheng  
School of Computing and Augmented Intelligence  
Arizona State University  
lcheng35@asu.edu

Mike Borowczak  
Department of Computer Science  
University of Wyoming  
mike.borowczak@uwyo.edu

ABSTRACT
Recent studies have shown Graph Neural Networks (GNNs) are extremely vulnerable to attribute inference attacks. To tackle this challenge, existing privacy-preserving GNNs research assumes that the sensitive attributes of all users are known beforehand. However, due to different privacy preferences, some users (i.e., private users) may prefer not to reveal sensitive information that others (i.e., non-private users) would not mind disclosing. For example, in social networks, male users are typically less sensitive to their age information than female users. The age disclosure of male users can lead to the age information of female users in the network exposed. This is partly because social media users are connected, the homophily property and message-passing mechanism of GNNs can exacerbate individual privacy leakage. In this work, we study a novel and practical problem of learning privacy-preserving GNNs with partially observed sensitive attributes.

In particular, we propose a novel privacy-preserving GCN model coined DP-GCN, which effectively protects private users’ sensitive information that has been revealed by non-private users in the same network. DP-GCN consists of two modules: First, Disentangled Representation Learning Module (DRL), which disentangles the original non-sensitive attributes into sensitive and non-sensitive latent representations that are orthogonal to each other. Second, Node Classification Module (NCL), which trains the GCN to classify unlabeled nodes in the graph with non-sensitive latent representations. Experimental results on five benchmark datasets demonstrate the effectiveness of DP-GCN in preserving private users’ sensitive information while maintaining high node classification accuracy.

CCS CONCEPTS
• Security and privacy → Privacy protections.

KEYWORDS
Graph Convolutional Network, Social Media, Privacy-Preserving, Disentangled Representation Learning, Orthogonal Subspace

1 INTRODUCTION
Graph Neural Networks (GNNs) have been widely used in various domains such as natural language understanding [33, 42], computer vision [34, 35], causal learning [5], social media mining [13], and recommendation system [43, 47]. GNN comes with both promises and perils. On one hand, it achieves remarkable performance in a wide range of applications; on the other hand, recent studies have shown that GNNs are vulnerable to attribute inference attacks [9, 12, 27]. For example, in a social network, an adversary can infer the sensitive attributes (e.g., race, age, and gender) of a user via different inference attack techniques such as using her behavioral record or information of her friends in a social network [15, 28]. Hence, developing privacy-preserving GNN models to resist inference attacks is of significant importance. Recent research efforts, such as studies using adversarial learning [27], federated learning [30, 52], differential private learning [44], have shown promising performance in individual privacy protection on graphs.

Nevertheless, existing privacy-preserving GNN models assume that sensitive attributes of all users are known beforehand. In practice, users have different privacy preferences (e.g., male users are typically less sensitive to their age information than female users), therefore, it should not be guaranteed that GNNs have access to all sensitive information. Take the scenario in Figure 1 as an example. There are six users in the social network. User 2, user 4, and user 6 are sensitive to their age and unwilling to reveal it. In contrast, user 1, user 3, and user 5 do not mind sharing their age information to make more friends. Here, we define the first type of users as private users (who are not willing to reveal their sensitive

Figure 1: Privacy problem with partially observed sensitive attributes in GNNs. Three private users (User 2, User 4, and User 6) treat age as their sensitive information and do not reveal it. However, the other three non-private users are willing to share their age information. Potentially, the age information of private users is leaked due to the homophily property and message-passing mechanism of GNNs.
attributes), and others as non-private users (who are willing to reveal their sensitive attributes). In this scenario, a GNN adversary can easily infer the age information of user 2, user 4, and user 6 using the observed age information of their neighbors. This is partly because the graph homophily property [29] and message-passing mechanism [23] of GNNs can exacerbate privacy leakage, leading to the age information of private users exposed. Generally, in homophilous graphs, nodes with similar sensitive attributes are more likely to connect to each other than nodes with different sensitive attributes [7]. For example, young people tend to make friends with people of similar age on the social network [8]. This phenomenon in Figure 1 severely violates individual privacy regulations. Thus, it is critical to investigate this novel and practical problem: learning privacy-preserving GNN with partially observed sensitive attributes.

To achieve both effective privacy-preserving performance and competitive performance in downstream tasks, we confront two major challenges: First, user dependency. Users in a graph are typically dependent to each other due to high connectedness. With the graph homophily property and message-passing mechanism of GNNs (We verified it in the experiment), a GNN adversary may infer a private user’s sensitive information through her/his non-private neighbors. Thus, we need to minimize the impact of the observed sensitive attributes of non-private neighbors on revealing the sensitive attributes of private users. Second, attribute dependency. Some non-sensitive attributes are naturally correlated with sensitive attributes [51]. For example, a user’s hobby may be related to her/his gender; zip code is often correlated with race. Previous studies [45, 49] have found that simply removing sensitive attributes still leads to privacy leakage due to the correlation between the non-sensitive and sensitive attributes. Therefore, we need to remove the hidden sensitive factors from non-sensitive attributes.

To address these challenges, we propose a principled privacy-preserving GCN model coined DP-GCN. It aims to effectively protect private users’ sensitive attributes when the sensitive attributes of non-private users are observed. The key is to remove the hidden sensitive factors in the original non-sensitive attributes such that only the non-sensitive factors are used in the downstream tasks. DP-GCN consists of two main modules: DRL module, Disentangled Representation Learning Module, disentangles the non-sensitive attributes of users’ into sensitive and non-sensitive representations in the latent space. To minimize the linear dependency between sensitive and non-sensitive latent representations, we further impose the orthogonal subspace projection [4]; NCL module, Node Classification based on Non-sensitive Latent Representation Module, aims to execute downstream tasks with the non-sensitive latent representations. In this paper, we use node classification task for illustration. We evaluate the proposed model on five benchmark datasets for node classification. The experimental results demonstrate the effectiveness of the proposed model in both privacy-preserving performance and node classification performance.

In summary, our main contributions in this work include:

1. We study a novel and practical problem of privacy-preserving GNNs with partially observed sensitive attributes.
2. We propose a principled privacy-preserving GNN model DP-GCN to mitigate the individual privacy leakage issue of private users. DP-GCN is designed based on disentangled representation learning with orthogonal constraint. We provide theoretical proof for the orthogonality of sensitive and non-sensitive representations.
3. We evaluate the proposed model on five benchmark datasets for node classification task and the experimental results show that our model achieves better privacy-preserving capability and competitive node classification performance simultaneously.

2 RELATED WORK

2.1 Privacy Attacks on GNNs

Different privacy attack techniques associated with GNNs have been proposed in recent years. Depending on the attacker’s goal, privacy attacks can be roughly categorized into four types: membership inference attack [32, 36], model extraction attack [21, 37, 39], link stealing attack [14, 50], and attribute inference attack [9, 11]. Membership inference attack aims to identify whether a node is used for training the target model. When a node is fully known to the adversary, it may directly lead to a privacy breach. For example, knowing that a certain patient’s clinical record was used to train a model associated with a disease can reveal that the patient had this disease [36]. Model extraction attack belongs to black-box privacy attacks. It seeks to extract information of model parameters and reconstruct one substitute model that behaves similarly to the target model. In link stealing attacks, the adversary has black-box access and can accurately infer whether there exists a link between any pair of nodes in a graph that is used to train the GNN models. By contrast, the goal of the adversary in attribute inference attack is to infer the sensitive attributes of an individual such as nationality, or political view. For instance, in social media, an attacker may infer a user’s gender accurately by tracking the list of pages that the user liked on Facebook. Such inferred attributes can be further leveraged to deliver customized advertisements to users [18]. Overall, these works underline various privacy risks associated with GNNs and demonstrate the vulnerability of GNN models to privacy attacks. In this work, we focus on attribute inference attacks on GNNs.

2.2 Privacy-Preserving GNNs

A straightforward solution to address node level privacy problem in GNNs is encryption [10]. However, encryption comes with an extra cost of time and computation. To address these challenges, one direction is mitigating privacy leakage via adversarial training. For example, Liao et al. [27] proposed an adversarial learning approach where they introduced a minimax game between the desired graph feature encoder and the worst-case attacker. Li et al. [25] presented a graph adversarial training framework that integrated disentangling and purging mechanisms to remove users’ private information from learned node representations. However, both of these works assume that the modeler for the downstream task has full access to the sensitive attributes, which may not be guaranteed in the real world. The second direction protecting node-level privacy in GNNs uses federated and split learning. In [52], the authors tackled the problem of privacy-preserving node classification by splitting the computation graph of a GNN among multiple data holders. They further used a trusted server to combine the information from different parties and complete the training process. Mei et al. [30] proposed to use structural similarity and federated learning to hide users’ private information in GNNs. Wu et al. [40] proposed
a federated GNN for privacy-preserving recommendation. However, these approaches rely on the existence of a trusted server for model aggregation. In practice, it is difficult to find a trusted party to store users’ sensitive information. The third direction leverages differential privacy to protect privacy in GNNs. For example, Sina et al. [31] proposed a node-level privacy-preserving GNN using local differential privacy [22]. Xu et al. [44] proposed a differentially private graphing embedding method by applying the objective perturbation on the loss function. However, both approaches need to introduce extra noise to the model, leading to decreased model utility. While existing privacy-preserving approaches are promising, they assume that users’ sensitive attributes are accessible before the model training process. In practice, this assumption may violate privacy regulations and users’ unwillingness to share private information (or users’ different privacy preferences). Dai et al. [7] studied a similar problem setting to ours. Their goal is to achieve model fairness rather than preserve privacy in GNNs. Fairness task aims to mitigate model bias w.r.t. certain sensitive attributes whilst privacy task protects users’ sensitive attributes. They showed an inspiring finding that GNN can achieve fairness with limited accessible sensitive attributes of non-private users as they can infer the unknown sensitive attributes of private users. This suggests a privacy breach in GNNs, i.e., an attacker can infer private users’ undisclosed sensitive attributes from partially observed sensitive attributes of non-private users in the same network.

Therefore, this work complements established research by considering that only non-private users disclose sensitive information. We aim to mitigate privacy leakage of private users mainly caused by the homophily property and message-passing mechanism of GNNs. We propose a novel privacy-preserving GNN model based on disentangled representation learning [16] with orthogonal constraint. A similar work by Klein et al. [24] only considered independent data and directly adopted the Variational Autoencoders (VAE), which assumes isotropic Gaussian as latent prior. By contrast, we propose to introduce orthogonality to disentangle data representations into sensitive and non-sensitive factors. Our approach needs no additional assumptions. For simplicity, we use Graph Convolution Network (GCN) for illustration. Our method can also be extended to other GNN models.

3 PRELIMINARIES AND PROBLEM STATEMENT

3.1 Graph Convolutional Network

Let \( G = (V, E) \) denote a graph with a set of \( N \) nodes \( V = \{v_1, \cdots, v_N\} \), connected by a set of edges \( E \subseteq V \times V \). Let \( X \in \mathbb{R}^{N \times d} \) indicate the feature matrix with each row representing the \( d \)-dimensional feature vector of a single node. We denote node \( v_i \) as a tuple \((x_i, s_i, y_i)\), where \( x_i \in \mathbb{R}^{d-k} \) is a vector of non-sensitive features, \( s_i \in \mathbb{R}^k \) is a vector of sensitive features, and \( y_i \) is the ground-truth label. The adjacency matrix \( A \in \mathbb{R}^{N \times N} \) describes the graph structure of \( G \): \( A_{ij} = 1 \) if there is an edge \( e_{ij} \) between node \( v_i \) and node \( v_j \), and \( A_{ij} = 0 \) otherwise. Following the common node classification setting [46], only part of the nodes \( V_L = \{v_1, \cdots, v_l\} \) are associated with ground-truth labels \( y_L = \{y_1, \cdots, y_l\} \).

Given a set of labeled nodes \( \{(v_i, y_i), i = 1, \cdots, l\} \), the goal of GCN for node classification is to learn a mapping function \( f(\theta) : V_L \rightarrow y_L \) parameterized by \( \theta \). \( f(\theta) \) takes the input of the node features and edges, then outputs the corresponding labels. The objective function of GCN can be formulated as

\[
L_{GCN}(\theta, A, X, y_L) = \min_\theta \sum_{v_i \in V_L} l(f(X, A, \theta); y_i),
\]

where \( l(f(X, A, \theta), y_i) \) is a loss function that measures the difference between model prediction and true label, such as the cross-entropy loss for binary classification. For a two-layer GCN, \( f(\theta) \) with \( \theta = (W^{(0)}, W^{(1)}) \) is defined as

\[
f(X, A, \theta) = \text{softmax}(\tilde{A} \sigma(AXW^{(0)})W^{(1)}),
\]

where \( \tilde{A} = D^{-1/2}(A + I)D^{-1/2} \) and \( D \) is the diagonal matrix of \( A + I \). \( W^{(0)} \) and \( W^{(1)} \) are the learnable parameters. \( \sigma \) is the activation function such as ReLU. GCN defines the aggregation coefficients as the symmetrically normalized adjacency matrix \( A \) and these coefficients are shared across all layers. The aggregator of GCN is expressed as

\[
h_i^{(l+1)} = \sigma(\sum_{j \in N_i} \tilde{A}_{ij} h_j^{(l)} W^{(l)}),
\]

where \( h_j^{(l)} \) is the hidden representation of node \( j \) at layer \( l \) and \( N_i \) denotes the set of neighbors of node \( i \), including \( i \) itself.

3.2 Problem Statement

Suppose there is a social network \( G = (V, E) \) where part of the users have revealed their sensitive attributes \( s_j \) due to their different privacy preferences. In addition, some users’ node labels \( v_i \) are known whilst the rest are not. One of the downstream tasks is to learn the GCN parameter \( \theta \) in order to predict the labels of all unlabeled users. The challenge is that with the partially observed sensitive attributes, during the training process, an adversary may easily infer the unknown sensitive attributes which private users prefer not to reveal in social media. This is mainly because users in social networks \( G \) are not independent of each other. Therefore, to protect the sensitive attributes of private users, in this paper, we study the problem of how to learn users’ latent representations in privacy-preserving GCN with partially observed sensitive attributes:

**Definition 1 (Learning Privacy-Preserving GCN with Partially Observed Sensitive Attributes).** Given a network \( G = (V, E) \) with labels \( v_i \) and partially observed sensitive attributes \( s_j \) of non-private users, we aim to learn users’ latent representation \( \hat{X} \) which excludes users’ sensitive information and the GCN parameter \( \theta \) to classify unlabeled users accurately using \( \hat{X} \).

4 DP-GCN: THE PROPOSED FRAMEWORK

Users have different privacy preferences, i.e., some (i.e., non-private users) may prefer to reveal information that others (i.e., private users) consider sensitive such as gender or age. A critical question is how to protect the sensitive information of private users. This is challenging to answer because of the user dependency, i.e., users in a network are typically highly connected, and the attribute dependency, i.e., sensitive and non-sensitive attributes can be strongly correlated. The key is to disentangle the hidden sensitive factors from the non-sensitive attributes of users in a network. Here, we
propose a novel framework DP-GCN built upon disentangled representation learning [2] to learn a privacy-preserving GNN with partially observed sensitive attributes. DP-GCN includes two modules: DRL (Disentangled Representation Learning) and NCL (Node Classification based on Non-sensitive Latent Representation) as illustrated in Figure 2. DRL removes users’ sensitive factors by disentangling the original feature representations into sensitive and non-sensitive latent representations that are orthogonal to each other. NCL aims to execute downstream tasks based on non-sensitive latent representations.

4.1 Disentangled Representation Learning with Orthogonal Constraint

DRL seeks to remove the sensitive information from the original feature representation of each user for privacy-preserving. The main challenges in this task include the previously introduced User Dependency and Attribute Dependency.

4.1.1 Learning Orthogonal Subspace for Privacy-Preserving. To overcome the above two challenges, we assume linear dependency between sensitive and non-sensitive attributes exists, and hypothesize that the non-sensitive attributes can be decomposed into two linear independent factors that correspond to sensitive and non-sensitive factors, respectively. To this end, we propose to disentangle the non-sensitive attributes into the sensitive and non-sensitive parts that are orthogonal to each other. The core idea is to learn two orthogonal subspaces \( W_1 \) and \( W_2 \) such that the sensitive (non-sensitive) information in original non-sensitive attributes is projected into a sensitive (non-sensitive) latent space. The orthogonal constraint enables the non-sensitive latent representation to have minimal linear dependency with sensitive latent representation [26]. In Figure 3, we elaborate how to learn two orthogonal subspaces.

Given a non-private user \( v \in G \) described by a feature vector with six attributes \( \{x_1, s_1, x_2, s_2, x_3, s_3\} \) where \( \{x_1, x_2, x_3\} \) denotes a set of non-sensitive attributes and \( \{s_1, s_2, s_3\} \) denotes a set of sensitive attributes. We first decompose the original feature vector into a non-sensitive vector \( x = [x_1, x_2, x_3] \) and a sensitive vector \( s = [s_1, s_2, s_3] \). Since non-sensitive attributes may be correlated with sensitive information, then we disentangle the non-sensitive vector \( x \) into a sensitive latent representation and a non-sensitive latent representation. The goal is to make the non-sensitive latent representation is linear independent of \( v \)’s sensitive latent representation as much as possible. To achieve this, we learn two orthogonal projections \( W_1 \) and \( W_2 \) such that the user’s sensitive information and non-sensitive information in \( x \) are mapped to two orthogonal subspaces respectively. We prove that the sensitive and non-sensitive latent representations of \( x \) are orthogonal in Theorem 2 (See Appendix 7.1).

**THEOREM 2.** Let \( x^T = [x_1, \ldots, x_d] \) \((d \geq 2)\) be a non-sensitive feature vector, if \( W_1 \in \mathbb{R}^{d \times d} \) and \( W_2 \in \mathbb{R}^{d \times d} \) are two orthogonal subspaces, then \( W_1 x \) is orthogonal to \( W_2 x \), i.e., \( W_1 x \perp W_2 x \).

4.1.2 Objective Functions of Orthogonal Subspace Learning. To construct the two orthogonal subspaces \( W_1 \) and \( W_2 \) with partially observed sensitive attributes, we first learn \( W_1 \) based on the observed sensitive attributes \( S_k \) of \( k \) users. Let \( X_k \) denote the non-sensitive attribute matrix of \( k \) users in a graph, then \( X_k W_1 \) denotes the sensitive latent representation. To learn an accurate projection subspace \( W_1 \), we use the Frobenius norm \( \| \cdot \|_F^2 \) to minimize the distance between the sensitive latent representation and the observed sensitive attributes. The objective function can be defined as follows:

\[
L_{\text{sensitive}} = \min_{W_1} \|X_k W_1 - S_k\|_F^2.
\]

where \( W_1 \) is the learnable parameter. Equation (4) aims to learn a projection subspace \( W_1 \) such that the observed sensitive attributes \( S_k \) can be reconstructed from the non-sensitive attributes \( X_k \).

To learn the subspace \( W_2 \) orthogonal to \( W_1 \), i.e., \( W_2 \perp W_1 \), by the definition of orthogonal subspace [3], if two subspaces are orthogonal, then the inner product of any two vectors from the two subspaces should be zero. Therefore, we can simply incorporate the orthogonal constraint by adding a regularization term that enforces the product of the two subspace matrices to be zero matrix. As each column of \( W_1 \) and \( W_2 \) is a base vector of their basis, by defining
Equation (5), we enforce $W_1$ and $W_2$ to be orthogonal to each other:

$$L_{\text{orth}} = \min_{W_2} \|W_1^T W_2\|^2_F.$$  

(5)

With protected users’ privacy, we also need to maintain the down-stream classification performance. Given the observable non-sensitive attributes $x_i$ of user $\alpha_i$, $W_2x_i$ then represents the non-sensitive latent representation. We minimize loss on training set as follows

$$L_t = \min_{W_2} \sum_{i \in Y_L} l(f(x_i, W_2, \Theta), y_i, \Theta),$$  

(6)

where $l(\cdot, \cdot)$ is a loss function such as cross entropy.

Combining Equation (5) and Equation (6), the final objective function for learning $W_2$ is as follows:

$$L_{\text{non-sensitive}} = \min_{W_2} \beta L_{\text{orth}} + \gamma L_t$$

$$= \min_{W_2} \beta \|W_1^T W_2\|^2_F + \gamma \sum_{i \in Y_L} l(f(x_i, W_2, \Theta), y_i, \Theta),$$  

(7)

where $\beta$ and $\gamma$ are two hyperparameters that balance the orthogonality of two subspaces and the classification accuracy.

### 4.2 Node Classification based on Non-sensitive Latent Representation

NCL module aims to learn the GCN parameter $\Theta$ to classify the unlabeled users accurately based on the non-sensitive latent representations of users learned from DRL. Since users’ sensitive information has been disentangled from the non-sensitive latent representations through the orthogonal constraint, i.e., linear dependency relation between the sensitive and non-sensitive latent representations is removed, users’ privacy in terms of the sensitive attributes is protected.

Let $\bar{X}$ be the non-sensitive latent feature matrix with each row representing the latent feature vector of a user. We further update $\Theta$ of the GCN classifier by minimizing the following negative log likelihood loss:

$$L_{\text{GCN}}(\Theta, A, \bar{X}, y_L) = \min_{\Theta} \sum_{i \in Y_L} l(f(\Theta, \bar{X}, A), y_i).$$  

(8)

### 4.3 Overall Algorithm

Based on the designs of disentangled representation learning for privacy-preserving in section 4.1 and node classification based on non-sensitive latent representation in section 4.2, the final objective function of the proposed framework is given as

$$\arg \min_{W_1, W_2, \Theta} L = L_{\text{GCN}} + L_{\text{sensitive}} + L_{\text{non-sensitive}}$$

$$= L_{\text{GCN}} + \alpha \|X_\Theta W_1 - S_\Theta\|^2_F + \beta \|W_1^T W_2\|^2_F + \gamma L_t,$$  

(9)

where $\alpha$, $\beta$, and $\gamma$ are three hyperparameters used to control the contribution of each term. The first term of Equation (9) ensures an accurate GCN classifier; the second term learns an accurate sensitive latent representation; the third term enforces the orthogonality between $W_1$ and $W_2$; and the final term guarantees classification accuracy when learning subspace $W_2$.

The overall algorithm is summarized in Algorithm 1. Lines 3-6 implement the DRL module; lines 7-8 implement the NCL module.

### 4.4 Optimization Process

Jointly optimizing $\Theta$, $W_1$, and $W_2$ in Equation (9) is challenging. Thus, in this work, we use an alternating optimization schema [19] to iteratively update the three model parameters.

**Update $\Theta$.** To update $\Theta$, we fix $W_1$ and $W_2$, and remove terms that are irrelevant to $\Theta$. The objective function in Equation (9) reduces to Equation (8), which is a typical GCN optimization problem. We can learn $\Theta$ via stochastic gradient descent.

**Update $W_1$.** To update $W_1$, we fix $W_2$ and $\Theta$, then remove terms that are irrelevant to $W_1$. Equation (9) is reduced to the following:

$$\min_{W_1} \|X_\Theta W_1 - S_\Theta\|^2_F + \beta \|W_1^T W_2\|^2_F.$$  

(10)

**Update $W_2$.** Similarly, to update $W_2$, we fix $W_1$ and $\Theta$, then Equation (9) is reduced to the following:

$$\min_{W_2} \|W_1^T W_2\|^2_F + \gamma L_t.$$  

(11)

We adopt stochastic gradient descent algorithm to optimize both $W_1$ and $W_2$.

### 5 EXPERIMENTS

We evaluate the effectiveness of DP-GCN on five benchmark datasets by answering the following research questions:

- **RQ 1.** How does DP-GCN perform on node classification task compared to the state-of-the-art GCN models?
- **RQ 2.** How does DP-GCN perform on privacy-preserving task compared to the state-of-the-art GCN models?
- **RQ 3.** How do properties of DP-GCN affect its performance?

To answer RQ 1-2, we compare DP-GCN with strong baselines for node classification and privacy-preserving. To answer RQ 3, we conduct ablation study to examine the influence of different components in Equation (9) and understand how model hyperparameters affect the performance of DP-GCN. The implementation details can be found in Appendix 7.2.

#### 5.1 Experimental Setup

**Datasets description.** Following [1, 7], we validate DP-GCN on five benchmark datasets for node classification, including two datasets with social networks, i.e., Pokec-z and Pokec-n, and three ethical datasets constructed in [1], i.e., German credit, Recidivism,
and Credit defaulter. Specifically, Pokec-z dataset consists of 67,796 users and each user is described by 275 attributes. Pokec-n dataset includes 66,569 users and each user is associated with 264 attributes. Similar to [7], we treat the region as the sensitive attribute and users’ working field is the target label. German credit dataset contains 1,000 users and each user is described by 27 attributes. The sensitive attribute in this dataset is users’ gender and the target label is good or bad credit risks. Recidivism dataset contains 18,876 defendants who got released on bail at the U.S. state courts during 1990-2009 [20]. Each defendant is associated with 18 attributes. Race is treated as the sensitive attribute, and the target label is whether the defendant is likely to commit a violent crime or not. Credit defaulter dataset includes 30,000 individuals connected based on the similarity of their spending and payment patterns. Similar to [1], we consider age as the sensitive attribute and predict whether an individual will default on the credit card payment or not.

5.1.2 Training set, validation set, and test set splitting. For each dataset, we randomly select 50% samples as the training set, 30% samples as the validation set, and 20% samples as the test set. To generate partially observed sensitive attributes, for each training set, we randomly sample 30% nodes with known sensitive attributes and the sensitive attributes of the rest nodes are assumed unknown.

5.1.3 Baselines. To our knowledge, we are not aware of existing privacy-preserving GNNs for similar tasks as they assume all users’ sensitive attributes are known before model training [22, 25, 40, 52]. Therefore, we select four commonly-used GCN models, i.e., GCN [23], LA-GCN [48], RGCN [53], and SGCN [41], as the baselines. Each of these baselines needs to first infer the unknown sensitive attributes of nodes, then learn node representations for node classification. That is, the input layer up to the second last layer are used to learn node representations; the last layer is used to train a softmax classifier for node classification. For fair comparisons, our framework is also built upon these graph representation learning methods. Note that the compared methods only learn node representations and do not consider privacy-preserving whereas the proposed method considers both tasks simultaneously.

5.1.4 Parameter setting. We train DP-GCN and each baseline on the training set and tune hyperparameters to select the model with the minimal loss on the validation set. Then, we use the selected model to evaluate the performance on the test set. In parameter analysis (more details in Section 5.4), the default ranges for all hyperparameters $\alpha, \beta, \gamma$ are [0.1, 0.5, 1, 5, 10]. The projection matrices $W_1$ and $W_2$ are initialized with each element independently and identically distributed from Gaussian distribution $N(0, 1)$. The epoch numbers for model training and inference are set to 300 and 200, respectively. The learning rate is 0.001. For each set of experiments, we report the average performance of 10 runs along with standard deviations.

5.1.5 Evaluation metrics. Following [17, 19, 38], we use accuracy to evaluate both the node classification performance and the privacy-preserving performance. In the first task, higher accuracy is desired. By contrast, in the second task, lower accuracy is desired. Since the target label is users’ sensitive attribute in the second task, the better a model can infer the unknown sensitive attributes, the less privacy it is able to preserve.

5.2 Node Classification Task

To answer RQ. 1, we compare the node classification performance of DP-GCN with the four aforementioned GCNs. For each baseline, we first infer the unknown sensitive attributes and then classify the unlabeled nodes with inferred sensitive attributes. Table 1 presents the results for the five benchmark datasets.

We have the following observations: (1) The classification accuracy of DP-GCN is competitive compared with all baselines. Take RGCN as an example, our model shows higher classification accuracy w.r.t. most datasets (i.e., Pokec-z: 84.49 vs. 84.26; Pokec-n: 86.69 vs. 86.34; German: 72.10 vs. 70.70; Recidivism: 91.78 vs. 90.19; Credit: 78.78 vs. 79.74). (2) Compared with the standard GCN and LA-GCN, our model significantly improves privacy-preserving performance with a slight loss of classification accuracy. This finding is consistent across most datasets. Figure 4 presents the changes of privacy inference accuracy and classification accuracy. As we can see, compared with standard GCN, DP-GCN reduces the privacy inference accuracy by 5.44% while only increasing the classification loss by 0.03% on Pokec-n dataset. For Pokec-z data set, it reduces the privacy inference accuracy by 6.84% with the classification accuracy increased by 0.02%. Compared with LA-GCN, our model

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pokec-z</th>
<th>Pokec-n</th>
<th>German credit</th>
<th>Recidivism</th>
<th>Credit defaulter</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN [23]</td>
<td>84.47 ± 0.13</td>
<td>86.72 ± 0.43</td>
<td>72.80 ± 1.79</td>
<td>93.25 ± 0.25</td>
<td>80.44 ± 0.24</td>
</tr>
<tr>
<td>LA-GCN [48]</td>
<td>85.63 ± 0.53</td>
<td>87.62 ± 0.64</td>
<td>72.91 ± 1.68</td>
<td>92.13 ± 0.53</td>
<td>79.87 ± 0.44</td>
</tr>
<tr>
<td>RGCN [53]</td>
<td>84.26 ± 0.08</td>
<td>86.34 ± 0.12</td>
<td>70.70 ± 1.99</td>
<td>90.19 ± 0.59</td>
<td>79.74 ± 0.32</td>
</tr>
<tr>
<td>SGCN [41]</td>
<td>84.05 ± 0.32</td>
<td>86.39 ± 0.14</td>
<td>71.5 ± 3.38</td>
<td>89.96 ± 0.56</td>
<td>80.40 ± 0.29</td>
</tr>
<tr>
<td>DP-GCN</td>
<td>84.49 ± 0.11</td>
<td>86.69 ± 0.19</td>
<td>72.10 ± 2.56</td>
<td>91.78 ± 0.76</td>
<td>78.78 ± 0.91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pokec-z</th>
<th>Pokec-n</th>
<th>German credit</th>
<th>Recidivism</th>
<th>Credit defaulter</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN [23]</td>
<td>77.35 ± 1.29</td>
<td>77.19 ± 2.01</td>
<td>92.17 ± 1.93</td>
<td>58.55 ± 0.51</td>
<td>91.13 ± 0.42</td>
</tr>
<tr>
<td>LA-GCN [48]</td>
<td>78.21 ± 1.71</td>
<td>77.58 ± 2.09</td>
<td>93.14 ± 2.11</td>
<td>54.93 ± 1.46</td>
<td>91.08 ± 0.40</td>
</tr>
<tr>
<td>RGCN [53]</td>
<td>72.19 ± 0.39</td>
<td>71.14 ± 0.26</td>
<td>91.31 ± 2.69</td>
<td>54.95 ± 0.44</td>
<td>90.93 ± 0.33</td>
</tr>
<tr>
<td>SGCN [41]</td>
<td>79.31 ± 0.25</td>
<td>81.60 ± 0.37</td>
<td>92.57 ± 1.57</td>
<td>54.98 ± 0.57</td>
<td>90.85 ± 0.42</td>
</tr>
<tr>
<td>DP-GCN</td>
<td>72.06 ± 1.22</td>
<td>72.99 ± 2.09</td>
<td>87.26 ± 3.84</td>
<td>54.87 ± 0.97</td>
<td>90.62 ± 0.51</td>
</tr>
</tbody>
</table>
reduces the inference accuracy by 7.86% while only increasing the classification loss by 1.33% on Pokec-z dataset.

5.3 Privacy-Preserving Task
To answer RQ. 2, we further validate the privacy-preserving performance of DP-GCN by inferring private users’ sensitive attributes. Our inference setting is as follows: Assume a modeler acts as an adversary, and tries to infer the sensitive attributes of private users for malicious purposes. The information of the adversary can access is different for baselines and DP-GCN. For baselines, both users’ original non-sensitive attributes and partially observed sensitive attributes are known. By contrast, DP-GCN only knows non-sensitive latent representations and partially observed sensitive attributes. Therefore, this inference task is to treat the sensitive attributes as true labels and predict the unknown sensitive attributes of private users based on the accessible information. We fix the training data in each dataset and sample 30% nodes with known sensitive attributes and 70% nodes with unknown sensitive attributes. Then, based on the accessible information, we infer the unknown sensitive attributes of private users by using baselines and DP-GCN respectively. Table 2 shows the inference results for unknown sensitive attributes of private users.

We see that DP-GCN outperforms the compared models in privacy-preserving performance on most datasets. For example, compared with LA-GCN, our model decreases inference accuracy for sensitive information from 78.21 to 72.06 on Pokec-z dataset and from 93.14 to 87.26 on German credit dataset, respectively. Similar observations can be found on the other three datasets. We also observe that DP-GCN consistently outperforms standard GCN. For example, our model better preserves users’ privacy on Pokec-n and on Credit defaulter datasets (72.99 vs. 77.19; 90.62 vs. 91.13). Compared with RGCN and SGCN, DP-GCN also has lower inference accuracy on most datasets. Along with findings in Section 5.2, we can conclude that DP-GCN achieves competitive performance w.r.t. downstream tasks meanwhile largely protects sensitive information of private users, therefore, corroborating the effectiveness of our model.

5.4 Ablation Study
To understand how other factors affect the performance of DP-GCN, we conduct ablation study and answer the third question in this subsection. We first analyze how different components in the objective function (Equation (9)) influence the model performance, then analyze the impact of inferred sensitive attributes and size of observed sensitive attributes respectively.

5.4.1 Parameter Analysis. We vary the hyperparameters $\alpha, \beta, \gamma$ in the range $[0.1, 0.5, 1, 5, 10]$ where $\alpha$ is to control sensitive term, $\beta$ is to control orthogonality term, and $\gamma$ is to control classification loss term. We then observe the changes of node classification performance and privacy-preserving performance on four datasets, respectively. Based on the results shown in Figure 5, we make the following observations:

Observation 1: When $\beta$ and $\gamma$ are fixed, and $\alpha$ gradually increases from 0.1 to 10, the privacy inference accuracy decreases on most datasets whereas the performance of node classification is more stable w.r.t. different datasets. This finding implies that as $\alpha$ increases in a certain range, the model focuses more on the $L_{\text{sensitive}}$ term, it achieves better privacy-preserving performance.

Observation 2: When $\alpha$ and $\gamma$ are fixed, and $\beta$ gradually increases from 0.1 to 10, we see that both classification and privacy inference accuracy decreases consistently on all datasets. This result indicates that the orthogonality term $L_{\text{orth}}$ helps improve the privacy-preserving performance with the cost of classification loss.

Observation 3: When $\alpha$ and $\beta$ are fixed, and $\gamma$ gradually increases from 0.1 to 10, we observe that the classification performance slightly increases on most datasets. This is because the $L_{\text{loss}}$ term is responsible for classification accuracy. However, the privacy inference accuracy also slightly increases on most datasets, indicating that the model’s privacy-preserving performance decreases. The reason is that the model puts more emphasis on the $L_{\text{loss}}$ term when $\gamma$ is increasing.

5.4.2 Impact of Inferred Sensitive Attributes. We investigate if the inferred sensitive attributes will affect the classification accuracy. In particular, we use standard GCN to compare the following two scenarios: (1) Classify unlabeled nodes without sensitive attributes; (2) Classify unlabeled nodes with inferred sensitive attributes. Since our model does not use the inferred sensitive attributes for node classification, this set of experiments focuses on standard GCN. Table 3 presents the results on the five datasets.

We can see that when the sensitive attributes are inferred, the node classification accuracy is slightly lower on Pokec-z, Pokec-n, German, and Credit datasets. However, on Recidivism dataset, the classification accuracy with inferred sensitive attributes is slightly higher. This result implies that the impact of inferred sensitive attributes...
attributes on node classification depends on how much the sensitive attributes contribute to the classification.

5.4.3 Impact of Size of Observed Sensitive Attributes. To examine whether the size of observed sensitive attributes $k$ will affect the privacy-preserving performance, we study the relation between privacy inference accuracy and $k$ for DP-GCN. With the training set on each dataset fixed, we vary $k$ in $\{10\%, 30\%, 50\%, 70\%, 90\%\}$ and set $a = 1, \beta = 5, y = 1$. The reason we set $\beta$ a larger value is that the change of privacy-preserving performance is more comparable with the increased $k$.

We observe from Figure 6 that the privacy inference accuracy becomes larger (i.e., degraded privacy-preserving performance) with increased $k$ on most datasets. This is because the model has access to more sensitive information with increased $k$, therefore, better inferring users’ privacy. However, the overall performance trends on the five datasets are different. For example, for Pokec-n, the accuracy first decreases and then increases when $k > 0.7$; for German credit, the privacy inference accuracy increases most significantly when $k < 0.3$ and then only slowly increases after.

6 CONCLUSION AND FUTURE WORK

In this paper, we study a novel problem of privacy-preserving GNNs with partially observed sensitive attributes. We propose a principled privacy-preserving GNN model DP-GCN to mitigate the individual privacy leakage of private users due to the homophily property and message-passing mechanism of GNNs. DP-GCN is designed based on disentangled representation learning with orthogonal constraint. We theoretically prove that the sensitive latent representation is orthogonal to non-sensitive latent representation. Experimental results on five benchmark datasets demonstrate that the proposed model achieves better privacy-preserving capability and competitive node classification performance. As we only consider linear dependency of attributes for disentangled representation learning, one future direction is to investigate non-linear relationships of attributes in data. It is also important to study the generalizability of the privacy-preserving designs in the proposed model and testify DP-GCN with other popular GNN models. Another line of research can investigate the long-term impact of DP-GCN given the evolving nature of social networks [6].
7 APPENDIX

7.1 Proof of Theorem 2

Proof. Let $x_s = W_1 x$ and $x_{ns} = W_2 x$, by the definition of orthogonal vectors, then we need to show that the inner product of $x_s$ and $x_{ns}$ is zero. That is, let $x_s^T = [x_s^1, x_s^2, \cdots, x_s^d]$ and $x_{ns}^T = [x_{ns}^1, x_{ns}^2, \cdots, x_{ns}^d]$, then

$$x_s^T x_{ns} = x_s^1 x_{ns}^1 + x_s^2 x_{ns}^2 + \cdots + x_s^d x_{ns}^d = 0. \tag{12}$$

When $d = 2$, then Equation (12) can be reformulated as

$$x_1^1 x_{ns}^1 + x_2^1 x_{ns}^2 + + x_1^2 x_{ns}^2 = 0, \tag{13}$$

where $<.,.>$ denotes inner product of two vectors and $W^1_i$ denotes the $i$-th column of $W_r$ ($r = 1, 2$).

Since $W_1 \perp W_2$, then $< W_1^1, W_2^1 > = 0$, $< W_1^2, W_2^2 > = 0$, $< W_1^1, W_2^2 > = 0$, $< W_1^2, W_2^1 > = 0$. Equation (13) holds.

When $d = 3$, then Equation (12) can be reformulated as

$$x_1^1 x_{ns}^1 + \cdots + x_3^1 x_{ns}^3 + < W_1^1, W_2^1 > = 0. \tag{14}$$

Each term in the left-side of Equation (14) is 0. Hence, Equation (14) holds with $d = 3$.

By induction, when $d > 3$, then Equation (12) can be reformulated as

$$x_1^1 x_{ns}^1 + \cdots + x_d^1 x_{ns}^d + < W_1^1, W_2^1 > = 0. \tag{15}$$

The left-side of Equation (15) consists of $d(d + 1)/2$ terms and each term is 0. This completes the proof, i.e., $x_s \perp x_{ns}$. \hfill \Box

7.2 Reproducibility

In this subsection, we provide more details of the experimental settings for reproducibility purpose.

The proposed model was implemented in Python 3.8.5 and PyTorch 1.9.0. The implementation code is available at \footnote{https://github.com/HuiHu1/Privacy-Preserving-Graph-Convolutional-Network.} We selected five benchmark datasets, which can be downloaded at \footnote{https://github.com/EnyanDai/FairGNN/tree/main/dataset.} and at \footnote{https://github.com/chirag126/nifty/tree/main/dataset.}.

The parameter settings are presented in Table 4.

![Table 4: Parameter settings in DP-GCN and the compared models.](image)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>seed1</td>
<td>[100, 300]</td>
<td>[100, 300]</td>
<td>[100, 300]</td>
<td>[100, 300]</td>
<td>[100, 300]</td>
</tr>
<tr>
<td>seed2</td>
<td>[10, 100]</td>
<td>[10, 100]</td>
<td>[10, 100]</td>
<td>[10, 100]</td>
<td>[10, 100]</td>
</tr>
<tr>
<td>epoch1</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>epoch2</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>weight-decay</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
<td>1e-5</td>
</tr>
<tr>
<td>dropout</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>hidden</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>layer size</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>lr</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.2</td>
</tr>
<tr>
<td>lr-w1</td>
<td>0.001</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>lr-w2</td>
<td>0.001</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

\[1\] https://github.com/HuiHu1/Privacy-Preserving-Graph-Convolutional-Network.

\[2\] https://github.com/EnyanDai/FairGNN/tree/main/dataset.