Adversary for Social Good: Leveraging Attribute-Obfuscating Attack to Protect User Privacy on Social Networks

Xiaoting Li[1]*, Lingwei Chen[2]*, and Dinghao Wu[3]

[1] Visa Research, Palo Alto, CA, USA
[2] Wright State University, Dayton, OH, USA
[3] Pennsylvania State University, University Park, PA, USA
xiaotili@visa.com, lingwei.chen@wright.edu, dwu@psu.edu

Abstract. As social networks become indispensable for people's daily lives, inference attacks pose significant threat to users' privacy where attackers can infiltrate users' information and infer their private attributes. In particular, social networks are represented as graph-structured data, maintaining rich user activities and complex relationships among them. This enables attackers to deploy state-of-the-art graph neural networks (GNNs) to automate attribute inference attacks for users' privacy disclosure. To address this challenge, in this paper, we leverage the vulnerability of GNNs to adversarial attacks, and propose a new graph adversarial method, called Attribute-Obfuscating Attack (AttrOBF) to mislead GNNs into misclassification and thus protect user attribute privacy against GNN-based inference attacks on social networks. Different from the prior attacks using perturbations on graph structure or node features, AttrOBF provides a more practical formulation by obfuscating optimal training user attribute values, and also advances the attribute obfuscation by solving the unavailability issue of test attribute annotations, black-box setting, bi-level optimization, and non-differentiable obfuscating operation. We demonstrate the effectiveness of AttrOBF on user attribute obfuscation by extensive experiments over three real-world social network datasets. We believe our work yields great potential of applying adversarial attacks to attribute protection on social networks.

Keywords: Attribute privacy · Inference attack · Social networks · Graph adversarial attack · Attribute obfuscation.

1 Introduction

Social networks have emerged as an indispensable part of our daily lives, allowing us to conveniently share personal ideas for social engagements. Such an interactive environment generates a large amount of user-oriented data. Due to its accessibility and information richness, this data attracts attackers to disclose users' sensitive information and fulfill their malicious intents (e.g., unwanted

* Equal contribution. Work done while at PSU.

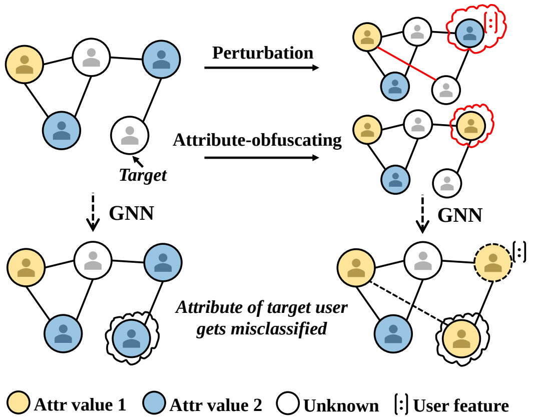


Fig. 1: GNN-based inference attack and graph adversarial attack leading to attribute obfuscation (attribute of target user gets misclassified) through traditional perturbation on graph structure/node feature or AttrOBF operation.

advertising, user tracing) [3, 35]. This puts users' privacy at risk. In fact, with the revolutionary development in machine learning, such privacy risk is not rare on social networks, and could be quickly transmitted and propagated through attribute inference attacks in an automatic fashion [9, 13, 16, 22, 26, 37].

In particular, social networks are naturally represented as graph-structured data, maintaining user activities and complex relationships among them. For example, nodes in these social graphs usually encode users' profiles, posts, photos, or other statuses, while edges connect users with their friendships, kinships, or follower-followee relationships. In the meanwhile, graph neural networks (GNNs) provide powerful techniques for graph understanding and mining [1, 15, 34, 19]. These GNNs take graph connectivity structure as filter to perform neighborhood information aggregation and extract high-level features from nodes and their neighborhoods [4], which have boosted the state-of-the-art for a variety of downstream tasks over graphs. Therefore, a surge of effective inference attacks utilize GNNs to reveal personal attributes (e.g., age, gender, location, career, and political views) that people are unwilling to disclose on social networks [7, 21, 32]. The idea is visualized as an example on the left-hand side of Figure 1 illustrating that the attribute of the target user can be correctly identified by leveraging GNNs over graph structure and user features.

In this work, we demonstrate an attribute privacy threat on social networks as the scenario that an attacker trains a well-performed GNN model to infer users' private attributes from graph-structured data such as Facebook friendship networks and Twitter follower-followee networks. With this in mind, some previous attempts have paid close attention to protect these attributes against inference attacks [3, 12, 16, 18, 25, 23, 27, 18], which, however, limit to unstructured image or text data [12, 23, 27, 18]. Thus, our goal here is to generalize the investigation to more challenging graph-structured data, and protect personal attribute privacy in this regard from a novel and practical adversarial learning perspective. Despite great success, recent studies [8, 31, 33, 38, 39, 5] have shown that GNNs remain vulnerable to adversarial attacks [6] that can easily fool the models into misclassification by performing small perturbations to graph structures and/or node features, which is shown in Figure 1. As the effectiveness of attribute inference attacks depends on high learning performance from GNN model while adversarial attacks substantially decrease its performance, this observation accordingly inspires us to take advantage of such a vulnerability and cast personal attribute privacy protection problem on social networks as an adversarial attack formulation problem against GNN-based attribute inference attacks. To achieve this goal, we face two challenges: (1) as inference attackers have a variety of choices in GNN construction, it is impossible for us to access the inference models for crafting graph adversarial attacks; (2) due to multimodality of user representations and intractability of relationship manipulations, modifications on either graph structures or node features cannot guarantee the validity of adversarial social networks, which are impractical in the real-world settings.

To address these challenges, in this paper, we design a black-box adversarial attack, called attribute-obfuscating attack (AttrOBF), which aims to deteriorate GNNs into misclassification and thus protect personal attribute privacy against GNN-based attribute inferences on social network data. Given a social network, AttrOBF proceeds by modifying a small fraction of optimal training users' attribute values, while the obfuscated attribute information can propagate along the whole graph through layer-wise neighborhood aggregations, such that the overall performance of attribute inferences by a surrogate GNN model is drastically degraded. Figure 1 illustrates the goal of our work. Due to transferability in adversarial machine learning [24], the obfuscated attribute over social networks is very likely to mislead the real attackers' inference GNN models. More importantly, it is necessary for inference attackers to collect initial attribute annotations for training, while users' annotating on social networks generally relies on their self-reporting; therefore, attribute obfuscating can be conveniently and proactively realized by users and data publishers, and also easily passed to subsequent inference attacks. These advantages allow a refined paradigm to efficiently mitigate the impacts of GNN-based inference attacks on attribute disclosure and enhance personal privacy protection in practice. In summary, our major contributions of this work are listed as follows:

- A novel and practical perspective of protecting privacy on social networks that leverages adversarial attacks to mitigate GNN-based inference attacks.
- A new adversarial attack AttrOBF for attribute obfuscation. To avoid NP-hard search, AttrOBF employs gradient-based method to obfuscate optimal training attribute values in an efficient way, where the problems regarding unavailability of test attribute annotations, black-box setting, bi-level optimization, and non-differential obfuscating operation are specially addressed.
- Extensive experiments on real-world social network datasets to evaluate the effectiveness of AttrOBF on attribute obfuscation and privacy protection.

2 Background and Related Work

2.1 Graph Neural Network for Attribute Inference

Social networks may indicate users' sensitive information, and thus easily expose them to the attackers who can access the data and infer the private attributes of interest to fulfill the economic, social, or political intents [27]. Considering that social networks are represented as graph-structured data, here we assume that the attackers would take advantage of user features and relationships to train GNN models so as to achieve their attribute inference goals [7, 21, 32].

Without loss of generality, we denote social network data G to be of the form $G = (V, E, \mathbf{X})$, where V ($n = |V|$) is the set of user nodes, E is the set of edges specifying relationships among users, and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is feature matrix. Nodes V can be further divided into annotated node set V_l ($n_l = |V_l|$) and unannotated node set V_u ($n_u = |V_u|$), where each annotated node is associated with a ground-truth attribute value $y \in Y = \{0, 1, \cdots, k-1\}$. For instance, for gender attribute, $Y = \{0\text{:male}, 1\text{:female}\}$. Edges E can be encoded as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_{ij} = \{0, 1\}$. That is, if $(v_i, v_j) \in E$, then $\mathbf{A}_{ij} = 1$; otherwise, $\mathbf{A}_{ij} = 0$. Given $\mathbf{A}$, $\mathbf{X}$, and V_l with attribute values $\mathbf{y}_l$, a GNN model $\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ ($\mathbf{Z} \in \mathbb{R}^{n\times k}$ and $k = |Y|$) is well trained to predict the attribute value for each node in V_u by minimizing the training loss as follows,

$$\mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}), \mathbf{y}_l) = \operatorname*{argmin}_{\mathbf{W}} l(\mathbf{Z}_l, \mathbf{y}_l) + \lambda\|\mathbf{W}\|_2^2 \tag{1}$$

where $\mathbf{W}$ is the trainable weight matrix, and $l(\cdot,\cdot)$ is the loss function. A GNN model $f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ can be specified as graph convolutional networks (GCNs) [15], graph attention networks (GATs) [28], or others [1, 10, 34]. GNNs can be applied under inductive and transductive settings. In this paper, we focus on transductive inferences where all node connections and features are accessible during training.

2.2 Graph Adversarial Attack for Attribute Protection

Given a private attribute, a graph adversarial attack attempts to perturb the graph to obfuscate that attribute and prevent GNN-based inference attack models from correctly identifying users' private attribute values. Generally, it modifies G with its structure and/or node features to an adversarial graph $\hat{G} = (\hat{\mathbf{A}}, \hat{\mathbf{X}})$ [8, 38, 39], such that the test loss over nodes in V_u can be maximized as follows,

$$\begin{aligned} &\max_{\hat{\mathbf{A}}, \hat{\mathbf{X}}} \mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\hat{\mathbf{A}}, \hat{\mathbf{X}}), \mathbf{y}_u) \\ &s.t.\ \mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\hat{\mathbf{A}}, \hat{\mathbf{X}}), \mathbf{y}_l),\ \|G - \hat{G}\|_0 \le \varDelta \end{aligned} \tag{2}$$

where a budget constraint $\varDelta$ is imposed on perturbations to limit the number of changes over node features and edges to ensure the imperceptibility of attacks.

Clearly, this is a challenging bi-level optimization problem: the attacker aims to maximize the test loss achieved after optimizing the model parameters on the modified graph $\hat{G}$; also, the action space of the attacker from G to $\hat{G}$ are discrete, enforcing vast combinatorial search [39]. Even worse, these attacks based on either graph structure or node feature manipulations are impractical in real-world social graph setting: (1) user nodes usually encode multi-modal data (e.g., profiles, posts, and other activities), where perturbations computed from the feature space are hard to map into user information space in an end-to-end manner; (2) due to limited access to large-scale social networks (especially for ones built on private interactions like Facebook), it is unreasonable to assume that users can alter any relationship as they wish. By contrast, users' attribute values can be easier to manipulate through users' self-reporting. It is necessary for inference attackers to collect initial attribute values for training, while these attribute values on social networks generally come from users' self-reporting. Therefore, attribute value manipulation has a direct impact on the model training and effectiveness of GNN-based inference attacks. Recent studies [20, 36] show that flipping a few training labels successfully drags down node classification accuracy to a great extent for graph models, which, however, can merely apply to binary classification tasks. To this end, in this paper, we would like to formulate a more general attribute-obfuscating method on social graphs to protect user attributes in practice, which specifically addresses the aforementioned challenges.

3 AttrOBF for User Privacy Protection

In this section, we first identify our goal and challenges, and then detail the technical steps of AttrOBF. The overview of AttrOBF is illustrated in Figure 2.

3.1 Attack Goal and Challenges

In our application setting, AttrOBF is designed to obfuscate a small fraction of optimal training users' attribute values so as to maximally decrease the overall performance of GNN-based attribute inferences trained on the modified graph. More specifically, given a target attribute with either binary or multiple classes, the goal is to have the test users classified as any attribute value different from the true one. In this regard, we can update the general graph adversarial attacks in Eq. (2), and the final objective function of AttrOBF has the following form.

$$\begin{aligned} &\min_{\Phi(\mathbf{y}_l)} -\mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\mathbf{A}, \mathbf{X}), \mathbf{y}_u) \\ &s.t.\ \mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}), \Phi(\mathbf{y}_l)),\ \ \|\Phi(\mathbf{y}_l) - \mathbf{y}_l\|_0 \le \epsilon n_l \end{aligned} \tag{3}$$

where $\Phi(\mathbf{y}_l)$ denotes the attribute obfuscating operation on the training attribute values $\mathbf{y}_l$, and ϵ is the obfuscating rate to n_l to ensure that AttrOBF is unnoticeable. Eq. (3) indicates the objective of AttrOBF that directly relates to the loss maximization on the test attribute values $\mathbf{y}_u$. Also, AttrOBF only performs changes to the training attribute values $\mathbf{y}_l$; hence we treat the graph structure $\mathbf{A}$ and node features $\mathbf{X}$ as two constants during our attack formulation. Eq. (3) poses four unique challenges to the design of our attack AttrOBF.

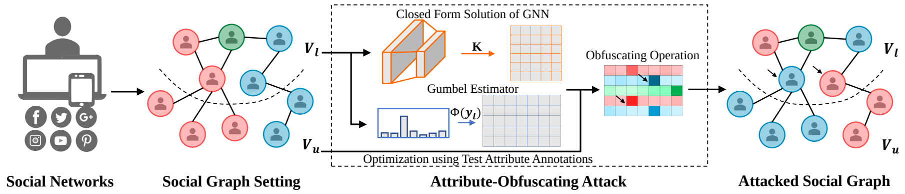


Fig. 2: The overview of AttrOBF to protect attribute privacy on social networks.

Black-box setting. AttrOBF is put under the black-box setting, where it is not aware of the GNN model $f_{\mathbf{W}}(\cdot, \cdot)$ used by inference attackers, including model choice, and parameters. As AttrOBF is a data poisoning attack while we aim to prevent inference attackers from disclosing users' private attribute values on our modified social networks, it is reasonable to assume that AttrOBF has access to the social graph data with respect to $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{y}_l$, which will be collected by inference attackers after attribute obfuscating in real-world scenarios.

Bi-level optimization. The problem formulation in Eq. (3) is of bi-level nature: the optimization on the attack loss $\mathcal{L}_{\text{atk}}$ is achieved after the optimization on the classification loss $\mathcal{L}_{\text{gnn}}$. In this respect, maximizing the test loss to obtain the optimal attribute obfuscating operation $\Phi(\mathbf{y}_l)$ requires retraining the GNN model, while the GNN model parameters $\mathbf{W}^*$ is constrained by the obfuscating operation $\Phi(\mathbf{y}_l)$ on the training attribute values. Optimizing such a bi-level problem is highly challenging by itself.

Non-differentiable obfuscating operation. In our graph setting, the training attribute data and the action space of the attribute obfuscating are discrete: the training attribute values are $\mathbf{y}_l = \{0, 1, \cdots, k-1\}^{n_l}$, and the possible actions are attribute value changes from the current one to any others. This makes the action space of the problem vast: given the maximum allowed training attribute value changes ϵn_l, the number of possible attacks is in $O((k-1)^{\epsilon n_l} n_l^{\epsilon n_l})$; exhaustive search is clearly infeasible, while greedy search easily leads to sub-optimal solution. Gradient-based methods can avoid the combinatorial search; however, discrete obfuscating operation $\Phi(\mathbf{y}_l)$ is non-differentiable, preventing AttrOBF from directly applying gradients to optimize the test loss.

3.2 Test Attribute Value Prediction

Transductive inferences over a graph imply that all node connections and features are accessible during training. Thus, we can use those annotated data to learn a GNN model described in Eq. (1) to estimate attribute values $\mathbf{y}_u$ of the unannotated or test nodes V_u

$$\mathbf{y}_u \approx \mathbf{y}_u^* = \operatorname*{argmax}_{i \in Y} \mathbf{Z}_{u,i}, \;\; \mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) \tag{4}$$

The advantage yielded here is that we can designate the surrogate model, which will be introduced in Section 3.3, as $f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ in Eq. (4) to estimate $\mathbf{y}_u$; if the adversarial attack formulated in a self-learning manner (i.e., using these predicted attribute values) has a high test error, it is very possible to also generalize poorly with the same surrogate model used to perform AttrOBF over the same graph. It is worth noting that only the attribute values $\mathbf{y}_l$ of the training nodes V_l are used to optimize the GNN model, while the test attribute annotations $\mathbf{y}_u$ from estimation are only used to maximize the test loss for attack formulation.

3.3 Surrogate Model

Under the black-box setting, we use two-layer Simple Graph Convolution (SGC) [30] as a surrogate model to perform our attribute-obfuscating attack on social graphs. Specifically, SGC is a linearized two-layer GCN

$$\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) = \text{softmax}(\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}),\ \mathbf{Z} \in \mathbb{R}^{n \times k} \tag{5}$$

where $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\mathbf{D}$ is the diagonal degree matrix defined on $\tilde{\mathbf{A}}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{n} \tilde{\mathbf{A}}_{ij}$.

There are three reasons behind this surrogate model choice: (1) SGC removes the non-linearity between GCN layers, which not only makes the model more tractable with less unnecessary complexity, but also captures the idea of graph convolutions (as demonstrated in [30], compared to those regular GNNs like GCN [15], GAT [28], FastGCN [4], SGC achieves the comparable or better test accuracy on different classification tasks); (2) SGC has been widely deployed as surrogate model in some successful graph adversarial attack formulations [36, 38, 39]; (3) SGC of linearity provides a simple closed form solution for $\mathbf{W}^*$, and thus transforms the bi-level optimization in Eq. (3) into single-level, which will be discussed in the following subsection. Due to transferability in adversarial machine learning [24], the attribute obfuscating operation optimized to mislead the surrogate model is very likely to degrade the real attackers' inference models.

3.4 Closed Form Solution

To solve the aforementioned bi-level optimization, nettack [38] trains a fixed surrogate model to reduce the attack to the problem simply built upon $\mathcal{L}_{\text{atk}}$; metattack [39] approximates the attack by choosing $\mathcal{L}_{\text{gnn}}$ as an alternate of $\mathcal{L}_{\text{atk}}$, arguing that a model of high training loss very likely misclassifies test nodes; some other attacks [20, 36] derive the model parameters and transform the bi-level optimization into single-level. Here, we leverage the closed form transformation idea to compute $\mathbf{W}^*$ and simplify the optimization on $\mathcal{L}_{\text{atk}}$.

Based on Eq. (1), Eq. (3), and Eq. (5), $\mathbf{W}^*$ can be rewritten as

$$\mathbf{W}^* = \underset{\mathbf{W}}{\text{argmin}}\ l((\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W}, \Phi(\mathbf{y}_l)) + \lambda\|\mathbf{W}\|_2^2 \tag{6}$$

After replacing the loss function $l(\cdot,\cdot)$ with mean square loss function, and considering attribute obfuscating operation $\Phi(\mathbf{y}_l)$ as an $n_l \times k$-dimensional matrix where each row is a one-hot vector specifying new attribute value, Eq. (6) can be further updated as

$$\mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \frac{1}{n_l} \|(\hat{\mathbf{A}}^2\mathbf{X})_l \mathbf{W} - \Phi(\mathbf{y}_l)\|_2^2 + \lambda \|\mathbf{W}\|_2^2 \tag{7}$$

In this way, we can approximately obtain the closed form of $\mathbf{W}^*$ through the derivation as follows,

$$\begin{aligned} &\frac{1}{n_l}\frac{\partial}{\partial \mathbf{W}}(\|(\hat{\mathbf{A}}^2\mathbf{X})_l \mathbf{W} - \Phi(\mathbf{y}_l)\|_2^2 + \lambda\|\mathbf{W}\|_2^2) = 0 \\ \Longrightarrow \quad & (\hat{\mathbf{A}}^2\mathbf{X})_l^T((\hat{\mathbf{A}}^2\mathbf{X})_l \mathbf{W} - \Phi(\mathbf{y}_l)) + \lambda \mathbf{W} = 0 \\ \Longrightarrow \quad & (\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l \mathbf{W} + \lambda \mathbf{W} = (\hat{\mathbf{A}}^2\mathbf{X})_l^T \Phi(\mathbf{y}_l) \\ \Longrightarrow \quad & \mathbf{W}^* = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda \mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T \Phi(\mathbf{y}_l) \\ \Longrightarrow \quad & \mathbf{W}^* = \mathbf{K}\Phi(\mathbf{y}_l) \end{aligned} \tag{8}$$

where we use $\mathbf{K} = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda \mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T$ for the sake of simplicity. Given the closed form of $\mathbf{W}^*$, the bi-level optimization of AttrOBF in Eq. (3) can be updated as the following single-level optimization on $\Phi(\mathbf{y}_l)$.

$$\begin{aligned} &\min_{\Phi(\mathbf{y}_l)} -\mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\mathbf{A}, \mathbf{X}), \mathbf{y}_u) \Rightarrow \\ &\min_{\Phi(\mathbf{y}_l)} -l((\hat{\mathbf{A}}^2\mathbf{X})_u \mathbf{K}\Phi(\mathbf{y}_l), \mathbf{y}_u) + \lambda\|\Phi(\mathbf{y}_l)\|_2^2 \\ s.t.\ & \|\Phi(\mathbf{y}_l) - \mathbf{y}_l\|_0 \le \epsilon n_l \end{aligned} \tag{9}$$

3.5 Gumbel Estimator

To solve the optimization problem in Eq. (9), the attribute obfuscating operation $\Phi(\mathbf{y}_l)$ is the key component. However, $\Phi(\mathbf{y}_l)$ is discrete thus non-differentiable, which means that we cannot directly use gradient-based methods to make updates on $\Phi(\mathbf{y}_l)$. To facilitate closed form solution in Section 3.4, we consider $\Phi(\mathbf{y}_l)$ as an $n_l \times k$-dimensional matrix, each row of which is represented as a one-hot vector to indicate the new attribute value. From the probabilistic perspective, we can model each attribute obfuscating operation as a categorical distribution, and this one-hot vector can be then sampled from k label probabilities $(p_0, p_1, \cdots, p_{k-1})$, where the position of 1 (i.e., the best obfuscating operation) is decided by the highest probability: $\texttt{one\_hot}(\text{argmax}_i[p_i])$.

In other words, given the categorical distribution $\mathbf{P} \in \mathbb{R}^{n_l \times k}$, the test loss of AttrOBF defined in Eq. (9) is an expectation over categorical variables.

$$\min_{\mathbf{P}} -\mathcal{L}_{\text{atk}}(\mathbf{P}) \Rightarrow \min_{\mathbf{P}} -\mathbb{E}_{\Phi(\mathbf{y}_l)\sim\mathbf{P}} l((\hat{\mathbf{A}}^2\mathbf{X})_u \mathbf{K}\Phi(\mathbf{y}_l), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2 \tag{10}$$

The categorical sampling $\Phi(\mathbf{y}_l) \sim \mathbf{P}$ is still non-differentiable. To solve Eq. (10), we need to find a good gradient estimator. To this end, we use Gumbel estimator [11] to draw samples $\Phi(\mathbf{y}_l)$ from $\mathbf{P}$ in a simple and efficient way. Different from

Algorithm 1: AttrOBF for attribute privacy protection.

```
Input: $G = (\mathbf{A}, \mathbf{X})$: Social graph $G$ with graph structure $\mathbf{A}$ and user features
       $\mathbf{X}$, $V_l$: $n_l$ training user nodes with attribute values $\mathbf{y}_l$, $V_u$: $n_u$ test user
       nodes without attribute values, $\epsilon$: obfuscating rate, $\tau$: temperature
       parameter, $T$: epochs.
Output: $\mathbf{y}_l$: the obfuscated training attribute values.
Train a GNN model using $\mathbf{A}$, $\mathbf{X}$ and $\mathbf{y}_l$ through Eq. (5);
Estimate $\mathbf{y}_u$ for the unannotated nodes $V_u$;
Pre-calculate $\hat{\mathbf{A}}^2\mathbf{X}$;
Pre-calculate $\mathbf{K} = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda\mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T$;
for each epoch $t \leq T$ do
    Sample $\mathbf{G} \sim \text{Gumbel}(0, 1)$;
    Calculate $h(\mathbf{P}, \mathbf{G})$ using Eq. (11);
    Calculate test loss $-\mathcal{L}_{\text{atk}}(\mathbf{P}) \approx -l((\hat{\mathbf{A}}^2\mathbf{X})_u\mathbf{K}h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2$;
    Update $\mathbf{P}$ by minimizing $-\mathcal{L}_{\text{atk}}(\mathbf{P})$;
end
$\Phi(\mathbf{y}_l) = \texttt{one\_hot}\ (\text{argmax}\ (\mathbf{P}, \text{axis} = 1))$;
Update $\mathbf{y}_l$ using new attribute values in $\Phi(\mathbf{y}_l)$ with top $\epsilon n_l$ highest
  probabilities in $\mathbf{P}$;
```

performing argmax to search for the maximal probability, the Gumbel estimator utilizes the Gumbel-Softmax function to generate continuous differentiable approximation to original categorical sampling. Specifically, let $\boldsymbol{\phi}$ (one row of $\Phi(\mathbf{y}_l)$) be sampled from the corresponding categorical distribution $\mathbf{p}$ (one row of $\mathbf{P}$); $\boldsymbol{\phi}$ is approximated as

$$\phi_i = h(\mathbf{p}, \mathbf{g}) = \frac{\exp\left((\log(p_i) + g_i)/\tau\right)}{\sum_{j=0}^{k-1} \exp\left((\log(p_j) + g_j)/\tau\right)}, \text{ for } i = 0, 1, \cdots, k-1 \tag{11}$$

where $\mathbf{g} \sim \text{Gumbel}(0, 1)$ is Gumbel distribution, and τ is the temperature controlling the steepness of softmax function. As the temperature increases, the expected value converges to a uniform distribution over the categories; on the contrary, as τ approaches 0, samples from the Gumbel-Softmax distribution become one-hot. Monte Carlo sampling from $\mathbf{g}$ makes Gumbel estimator unbiased and low variance [20]. Let $\mathbf{G} = [\mathbf{g}_0, ..., \mathbf{g}_{k-1}]^T$; by replacing $\Phi(\mathbf{y}_l)$ with $h(\mathbf{P}, \mathbf{G})$, the final test loss of AttrOBF is updated as

$$\min_{\mathbf{P}} -\mathcal{L}_{\text{atk}}(\mathbf{P}) \Rightarrow \min_{\mathbf{P}} -\mathbb{E}_{\mathbf{G}} l((\hat{\mathbf{A}}^2\mathbf{X})_u\mathbf{K}h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2 \tag{12}$$

Accordingly, the derivative of $-\mathcal{L}_{\text{atk}}(\mathbf{P})$ regarding the categorical distribution $\mathbf{P}$ can be computed in an approximate way.

$$-\frac{\partial\mathcal{L}_{\text{atk}}(\mathbf{P})}{\partial\mathbf{P}} \approx -\frac{\partial}{\partial\mathbf{P}}\left[l((\hat{\mathbf{A}}^2\mathbf{X})_u\mathbf{K}h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2\right] \tag{13}$$

The problem in Eq. (13) is differentiable and tractable. Therefore, it can be easily solved by gradient-based methods (e.g., stochastic gradient descent, Adam).

After the categorical distribution $\mathbf{P}$ is optimally updated, the attribute obfuscating operation $\Phi(\mathbf{y}_l)$ is uniquely defined as:

$$\Phi(\mathbf{y}_l) = \texttt{one\_hot}\ (\text{argmax}\ (\mathbf{P}, \text{axis} = 1)) \tag{14}$$

Note that, $\Phi(\mathbf{y}_l)$ indicates the obfuscating operation on the whole training attribute values $\mathbf{y}_l$. As specified in Eq. (3) and Eq. (9), to ensure the imperceptibility of attack, the attribute obfuscating operation is constrained by $\|\Phi(\mathbf{y}_l) - \mathbf{y}_l\|_0 \leq \epsilon n_l$. That is, the number of maximum allowed training attribute value changes is ϵn_l. As such, we leverage $\Phi(\mathbf{y}_l)$ and $\mathbf{P}$ to decide the actual attribute obfuscating: we first collect all new training attribute values from $\Phi(\mathbf{y}_l)$ that are different from the original and their corresponding probabilities from $\mathbf{P}$, and then use those new attribute values with top ϵn_l highest probabilities to update $\mathbf{y}_l$ so as to guarantee the optimal operation. Algorithm 1 illustrates our proposed attribute-obfuscating attack AttrOBF to protect attribute privacy on social networks. As graph structure $\mathbf{A}$ and node features $\mathbf{X}$ are constants during attribute-obfuscating attack, we can pre-calculate $\hat{\mathbf{A}}^2\mathbf{X}$ and $\mathbf{K}$ using $O(\max(n^3, d^3))$, which significantly decreases the time complexity for each optimization iteration to $O(n_l n_u d)$ $(k \ll d)$. Therefore, this efficient attack strategy has implications on its applicability for attribute protection on large social networks in practice.

4 Experimental Results and Analysis

4.1 Experimental Setup

Datasets. In our practical setting, we utilize three real-world social network datasets to conduct our experiments: Polblogs [2], Yale [17], and Rochester [17]. Polblogs represents a political blog network where their attribute values indicate political view of each user. Yale and Rochester datasets collect all the Facebook friendships of Yale University and Rochester University as well as some user attributes, in which career, gender, class year serve as private attributes. We train GNN models in a standard transductive setting where all node features are utilized and 20 nodes are annotated per class, and another 500 annotated nodes are viewed as validation set. Then, we randomly sample 1,000 nodes to evaluate the performance. Table 1 presents the dataset statistics.

Baseline methods and parameter settings. In our study, the proposed AttrOBF is designed for practical attribute privacy protections in social media, and to the best of our knowledge, graph adversarial attacks via modifications on multi-class annotations have not yet been explored. Thus, we formulate a couple of baselines in this regard to compare against AttrOBF: (1) Random attribute-obfuscating attacks (**Rand-obf**) where we randomly select a number of training nodes and obfuscate their attribute values to a random one. (2) Degree-based attribute-obfuscating attacks (**Deg-obf**) where we obfuscate the training nodes with the highest degrees because we believe these nodes play a more important role in the information propagation for GNNs than those with lower degrees;

Table 1: Statistics of three social network datasets in five attribute settings.

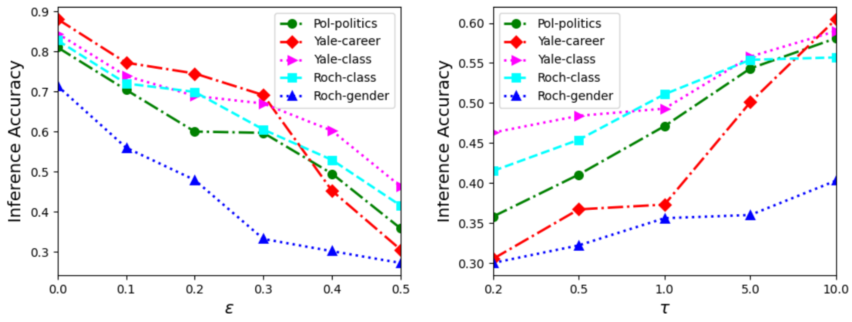
Dataset	**Attr.**	**Nodes**	**Edges**	**Classes**	**Train./Val./Test**
Polblogs	Politics	1,490	19,025	2	40/500/950
Yale	Career	8,578	405,450	2	20 × classes/500/1000
	Class-year			6	
Rochester	Gender	4,563	167,653	2	20 × classes/500/1000
	Class-year			5	

similarly, for all inference settings, we modify the attribute values of the selected nodes to a random one. Note that, as we only focus on attribute obfuscating, those adversarial methods designed for different settings, such as manipulating graph structure or node features, are not comparable here. Following the baseline designs in [36], in order to investigate how different components affect the performance of AttrOBF, we further formulate two variants as baselines by replacing surrogate model and loss function: (3) **AttrOBF-lp** follows the same steps of AttrOBF except that we use label propagation as our surrogate model, which accordingly updates the closed form in Eq. (8) and single-level optimization in (9). (4) **AttrOBF-cse** replaces mean square error in loss function to cross-entropy, which updates the final test loss of AttrOBF in Eq. (12). In our parameter settings, we set the optimization epoch in AttrOBF as 1,000 and training epoch of GNN models as 200. The temperature parameter for Gumbel estimator τ introduced in Eq. (11) is set as 0.2 and $\lambda = 0.01$ for optimization.

Attack model for attribute inference attacks. Attackers conduct attribute inference attacks to disclose private attributes of users by learning a GNN model on public social network data. Since we do not know the attacker's model, we use SGC to solve black-box setting and closed form for AttrOBF. In our experimental setting, we train simple graph convolution (SGC) [30], graph convolutional network (GCN) [15], graph attention network (GAT) [28], and GCN-based label propagation network (GCN-LP) [29] to perform the inference attack. We mainly use GCN to evaluate the effectiveness of AttrOBF and the impacts of different parameters, while the comparisons among these four models are leveraged for transferability evaluation in Section 4.4. To be comparable, these four GNN models are of two-layer structure and the dimension of the hidden layer is set as 16. All other model parameters align with their original works [30, 15, 28, 29].

4.2 Evaluation of AttrOBF

Effectiveness. In our experiments, we test the results of five inference settings (i.e., Polblogs-politics, Yale-career, Yale-class, Rochester-class, Rochester-gender) while using AttrOBF to obfuscate the training attribute values with obfuscating rate $\epsilon \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, where 0.0 means no attack in place. It is worth mentioning that we merely modify 10 training nodes per class even when reaching the largest obfuscating rate 0.5. We believe this complies



(a) (b)

Fig. 3: (a) represents the test accuracy of all inference tasks on different attribute obfuscating rate ϵ, while (b) specifies the evaluation results of AttrOBF under different values of temperature parameter τ.

with its practicability requirement considering the large graph volume. In this experiment, we use test accuracy to evaluate attribute privacy protection performance. The lower test accuracy represents the better performance of our method. The experimental results are shown in Figure 3(a). We can see that the attribute inference accuracy for Polblogs-politics, Yale-career, Yale-class, Rochester-class and Rochester-gender on clean data is 81.1%, 88.1%, 84.5%, 82.8%, and 71.4%, which are relatively close to the state-of-the-art results on each dataset. Obviously, AttrOBF drastically decreases all the accuracy of inference attacks and thus achieves the goal of protecting users' attribute privacy on social networks.

Impact of attribute obfuscating rate ϵ. Intuitively, when we enlarge the ϵ, the number of the training node attribute values obfuscated by AttrOBF increases and the accuracy of inference attacks should decrease. The results in Figure 3(a) confirm this point: as the obfuscating rate increases from 0.0 to 0.5, the inference accuracy drops 45.3% for Polblogs-politics, 57.6% for Yale-career, 41.2% for Yale-class, 41.3% for Rochester-class, and 44.2% for Rochester-gender. We can also observe that AttrOBF obtains better performance on binary inference settings than multi-class inference tasks. The reason behind this could be that attacking space on multi-class social graphs is larger, which leads to more uncertainty and difficulty than binary problems that flipping labels can directly impact on neighborhoods and thus more easily mislead the GNN model.

Impact of temperature for Gumbel estimator τ. The temperature τ for Gumber estimator is an important parameter in our method that controls the effectiveness of the one-hot sampling. We gradually increase the value of τ in AttrOBF to analyze its impact to the attack performance. In the experiments, we assess the effectiveness of AttrOBF with temperature $\tau \in \{0.2, 0.5, 1.0, 5.0, 10.0\}$ in five inference settings when $\epsilon = 0.5$. We show the results in Figure 3(b). We can see from the figure that AttrOBF achieves the best performance when

Table 2: Inference accuracy of using true or estimated test attributes.

Test labels	Pol-politics	Yale-career	Yale-class	Roch-class	Roch-gender
True	33.1%	29.3%	43.0%	40.9%	25.7%
Estimated	35.7%	30.5%	43.3%	41.5%	27.1%

$\tau = 0.2$ for all inference tasks. As τ increases, the capability of our adversarial attack in alleviating the inference models is degraded. This is because when we continuously amplify the τ value, Gumbel-Softmax distribution becomes closer to uniform distribution, which more significantly deviates from one-hot sampling and thus affects the effectiveness of attribute obfuscating operation. There is a trade-off between near-zero temperatures, where samples are identical to one-hot but the variance of the gradients is large as well. Based on this fact, we use $\tau = 0.2$ throughout the following evaluations.

Impact of test attribute annotations $\mathbf{y}_u$. We use the prediction results of the surrogate model to estimate the test attribute values in our evaluations, and compare with the true test attribute annotations to investigate the impact of them on the performance of AttrOBF. The comparative results are shown in Table 2 with obfuscating rate $\epsilon = 0.5$. We can observe that integrating true test attribute annotations in our objective loss function can obtain better attack results than the estimated ones, as the estimation might introduce extra loss in our objectives. However, the inference accuracy difference between using true and estimated test attribute annotations seems not very significant. The reason behind this could be that the surrogate model's inference accuracy for different attribute settings is relatively high (i.e., 81.1%, 88.1%, 84.5%, 82.8%, and 71.4% for Polblogs-politics, Yale-career, Yale-class, Rochester-class and Rochester-gender respectively), which makes the estimation closer to ground truth. This implies that our method is not tightly coupled with true test attribute annotations, and can be easily feasible in practical applications.

4.3 Comparisons with Other Attack Baselines

In this section, we compare our method AttrOBF against four baselines: Rand-obf, Deg-obf, AttrOBF-lp and AttrOBF-cse. For all methods, we set the obfuscating rate ϵ as 0.5, and use GCNs as the attack model to assess the inference accuracy. The results of five inference settings are presented in Table 3. We can observe that our method AttrOBF significantly outperforms Rand-obf on all inference tasks. Under Rand-obf attack, the inference accuracy only slightly decreases for all obfuscating rates, which indicates that GCNs are quite robust to random label noise. This also benefits from the powerful learning capability of GCNs on graph data of embracing both node features and graph topological structure. Therefore, GCNs are resilient against random node obfuscating operations but still vulnerable to our well-designed adversarial attacks. AttrOBF also achieves better performance than Deg-obf attack, especially for multi-class

Table 3: Comparisons with other attack baselines (inference accuracy).

Setting	Rand-obf	Deg-obf	AttrOBF-lp	AttrOBF-cse	AttrOBF
Pol-politics	55.7%	37.0%	42.5%	36.5%	**35.7%**
Yale-career	61.2%	47.2%	49.4%	38.6%	**30.5%**
Yale-class	72.0%	53.1%	45.5%	43.8%	**43.3%**
Roch-class	69.6%	54.2%	43.5%	42.1%	**41.5%**
Roch-gender	46.7%	42.1%	39.9%	31.0%	**27.1%**

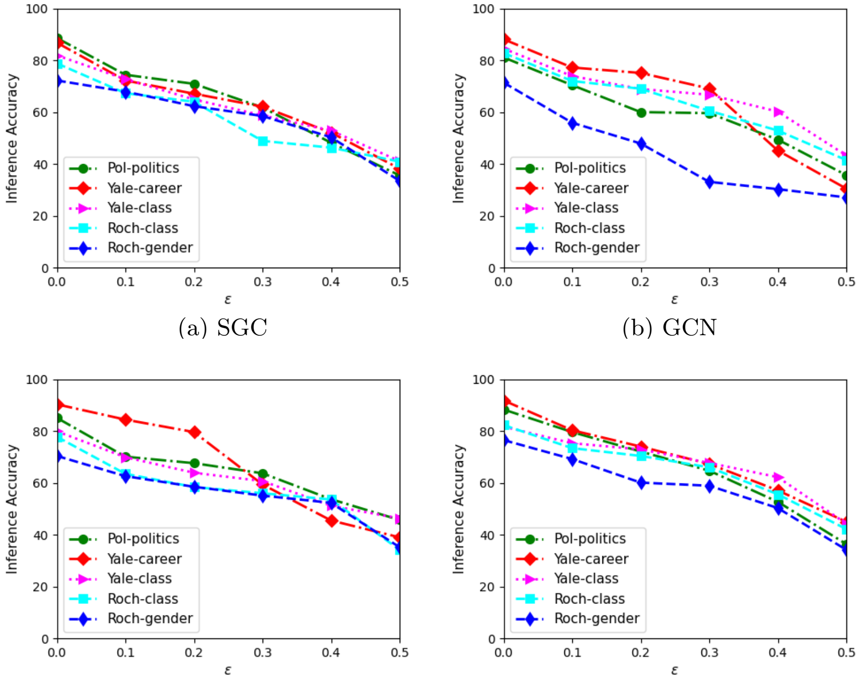
inference problems. For instance, AttrOBF reduces the inference accuracy to 43.3% and 41.5% for Yale-class and Rochester-class while the results of Deg-obf attack are 53.1% and 54.2%, respectively. This is due to the fact that adversarial attribute values generated by AttrOBF are specifically derived from the goal of misleading the learning model, which are much more effective to degrade the performance of node classification, while Deg-obf identifies the degree information of nodes as the only influential factor for graph learning but ignores other conditions (e.g., node features) leveraged by GCNs.

Regarding to AttrOBF-lp, AttrOBF achieves better results for all classification settings. Compared to graph neural networks, label propagation only aggregates the label information from nodes' neighbors without considering the important feature information. Therefore, choosing SGC to be the surrogate model to compute the closed form solution is more reasonable and effective for our formulation. The similar variant AttrOBF-cse can achieve comparable results but still slightly underperforms our method. The reason behind this performance difference could be that mean square error can better formalize the discrepancy between ground truth and prediction results in the embedding space.

4.4 Transferability of AttrOBF

Under the black-box setting, we don't know what model the attacker is using to infer private attributes. This naturally leads us to the question: can our attack strategy generalize to other inference attack models? To answer this question, in this evaluation, we explore the transferability of our method AttrOBF. Specifically, we deploy AttrOBF to obfuscate the training attribute values and generate adversarial graph on five attribute inference settings. Then we test the inference results of the poisoned data against four state-of-the-art GNN models, including SGC [30], GCN [15], GAT [28] and GCN-LP [29] under five obfuscating rates (i.e., $\epsilon = \{0.1, 0.2, 0.3, 0.4, 0.5\}$). To ensure our results are comparable, we build up these models with the same parameter and data settings.

The results presented in Figure 4 show that the adversarial attack performed by AttrOBF can successfully transfer to different graph neural networks. Our AttrOBF method learned on a linearized GCN (i.e., SGC) presents the similar effectiveness against different GNN models under the same inference setting. For example, when ϵ is set as 0.5, AttrOBF reduces the accuracy of SGC, GCN, GCN-LP to 35.6%, 35.7% and 36.4% on polblogs-politics inference attack and



(a) SGC

(b) GCN

(c) GAT

(d) GCN-lp

Fig. 4: (a), (b), (c) and (d) specify the inference accuracy of SGC, GCN, GAT and GCN-lp while conducting AttrOBF on our surrogate model over different data settings; lower inference accuracy indicates better attack transferability.

33.5%, 27.1% and 34.2% on Rochester-gender inference setting. For Yale-career, the inference accuracy of all GNN models drops over 30% when increasing ϵ from 0.1 to 0.5. While for Yale-class and Rochester-class inference settings, the transferability of AttrOBF on four GNN models are very close and slightly underperform other inference tasks. On the other hand, the results also imply that the complexity of the surrogate model and the intrinsic adversarial vulnerability of the target model contribute to attack transferability: the attack results on SGC and GCN outperform those with more complex model structure such as GAT and GCN-LP. Since the target models are uncontrollable, when applying AttrOBF in practice, we may need to elaborate the surrogate model for better transferability. We leave it as our future exploration.

5 Impact, Applicability and Limitation

Our previous method formulation and experimental evaluations demonstrate the impact of our proposed graph adversarial attack solution for attribute privacy protection on social networks: (1) as graph structure and node features are not perturbed, the utilities of social networks regarding user activities and relationships are well preserved without any influence on other downstream tasks; (2) mere small yet optimal training annotation changes can effectively mitigate attribute inference attacks; (3) attribute obfuscating is easy to operate for both data publishers and users. Therefore, in practice, AttrOBF can work as an easy-to-use API provided on the social network server side that enables data publishers to either locally or globally manipulate user attribute values before making the social graphs publicly available, or warn users of potential attribute privacy threats such that users can proactively change their attribute information on the client side. Nonetheless, our approach also poses a limitation which we discuss as follows. In our experiments, we train some regular GNN-based attack models for attribute inferences on social networks. Though AttrOBF has been validated to be transferable to these GNNs, the attackers could take advantage of more advanced and robust GNN models (e.g., adversarial training via latent perturbation [14]) to infer attributes and thus deteriorate AttrOBF. We acknowledge this limitation and leave the investigation on this arms race as our future work, yet it does not impact the great value and general validity of our new insight about leveraging graph adversarial attacks for attribute obfuscation and privacy protection on social networks in practice, as graph learning models of inherent vulnerability could always be evaded by more complicated and more sophisticated adversarial techniques.

6 Conclusion

In this paper, we investigate adversary for social good, and cast attribute privacy protection problem on social networks as a graph adversarial attack formulation problem to defend against GNN-based attribute inference attacks. We design a black-box attribute-obfuscating attack AttrOBF, where a linearized two-layer GCN is used as a surrogate model to perform our attack. Under the help of this surrogate model, a closed form of model weights is obtained to transform the bi-level optimization for AttrOBF into single-level. To address non-differentiable attribute obfuscating operation, we introduce Gumbel estimator to generate continuous differentiable approximation that enables gradient-based methods to search for the optimal training attribute values to change. We conduct extensive experimental studies on real-world social network datasets to evaluate the performance of AttrOBF, which validate its effectiveness against GNN-based attribute inference attacks. Despite the limitation, we believe that our work has implications on the applicability of adversarial attacks for attribute obfuscation and privacy protection in practice.

Acknowledgement

The work was supported in part by a seed grant from the Penn State Center for Security Research and Education (CSRE).

References

1. Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., Galstyan, A.: Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: International Conference on Machine Learning. pp. 21–29 (2019)
2. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 us election: divided they blog. In: Proceedings of the 3rd international workshop on Link discovery. pp. 36–43 (2005)
3. Beigi, G., Shu, K., Zhang, Y., Liu, H.: Securing social media user data: An adversarial approach. In: Hypertext and Social Media. pp. 165–173 (2018)
4. Chen, J., Ma, T., Xiao, C.: Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247 (2018)
5. Chen, L., Li, X., Wu, D.: Enhancing robustness of graph convolutional networks via dropping graph connections. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 412–428 (2020)
6. Chen, L., Ye, Y., Bourlai, T.: Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In: 2017 European Intelligence and Security Informatics Conference (EISIC). pp. 99–106. IEEE (2017)
7. Chen, W., Gu, Y., Ren, Z., He, X., Xie, H., Guo, T., Yin, D., Zhang, Y.: Semi-supervised user profiling with heterogeneous graph attention networks. In: IJCAI. vol. 19, pp. 2116–2122 (2019)
8. Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., Song, L.: Adversarial attack on graph structured data. arXiv preprint arXiv:1806.02371 (2018)
9. Gong, N.Z., Liu, B.: Attribute inference attacks in online social networks. TOPS **21**(1), 3 (2018)
10. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216 (2017)
11. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
12. Jia, J., Gong, N.Z.: Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In: USENIX Security. pp. 513–529 (2018)
13. Jia, J., Wang, B., Zhang, L., Gong, N.Z.: Attriinfer: Inferring user attributes in online social networks using markov random fields. In: WWW. pp. 1561–1569 (2017)
14. Jin, H., Zhang, X.: Robust training of graph convolutional networks via latent perturbation. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III. pp. 394–411 (2021)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
16. Kumar, C., Ryan, R., Shao, M.: Adversary for social good: Protecting familial privacy through joint adversarial attacks. In: AAAI (2020)
17. Li, K., Luo, G., Ye, Y., Li, W., Ji, S., Cai, Z.: Adversarial privacy preserving graph embedding against inference attack. IEEE Internet of Things Journal (2020)
18. Li, X., Chen, L., Wu, D.: Turning attacks into protection: Social media privacy protection using adversarial attacks. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). pp. 208–216. SIAM (2021)
19. Liu, S., Chen, L., Dong, H., Wang, Z., Wu, D., Huang, Z.: Higher-order weighted graph convolutional networks. arXiv preprint arXiv:1911.04129 (2019)

20. Liu, X., Si, S., Zhu, X., Li, Y., Hsieh, C.J.: A unified framework for data poisoning attack to graph-based semi-supervised learning. arXiv:1910.14147 (2019)
21. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14424–14432 (2020)
22. Morgan-Lopez, A.A., Kim, A.E., Chew, R.F., Ruddle, P.: Predicting age groups of twitter users based on language and metadata features. PloS one **12**(8) (2017)
23. Oh, S.J., Fritz, M., Schiele, B.: Adversarial image perturbation for privacy protection a game theory perspective. In: ICCV. pp. 1491–1500 (2017)
24. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016)
25. Qian, J., Li, X.Y., Jung, T., Fan, Y., Wang, Y., Tang, S.: Social network de-anonymization: More adversarial knowledge, more users re-identified? TOIT **19**(3), 1–22 (2019)
26. Ruder, S., Ghaffari, P., Breslin, J.G.: Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. arXiv preprint arXiv:1609.06686 (2016)
27. Shetty, R., Schiele, B., Fritz, M.: A4nt: author attribute anonymity by adversarial training of neural machine translation. In: Proceedings of the 27th USENIX Security Symposium (USENIX Security 18). pp. 1633–1650 (2018)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
29. Wang, H., Leskovec, J.: Unifying graph convolutional neural networks and label propagation. arXiv preprint arXiv:2002.06755 (2020)
30. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. PMLR (2019)
31. Wu, H., Wang, C., Tyshetskiy, Y., Docherty, A., Lu, K., Zhu, L.: Adversarial examples for graph data: Deep insights into attack and defense. In: IJCAI. pp. 4816–4823 (2019)
32. Wu, Y., Lian, D., Jin, S., Chen, E.: Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In: IJCAI. pp. 3898–3904 (2019)
33. Xu, K., Chen, H., Liu, S., Chen, P.Y., Weng, T.W., Hong, M., Lin, X.: Topology attack and defense for graph neural networks: An optimization perspective. arXiv preprint arXiv:1906.04214 (2019)
34. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: SIGKDD. pp. 974–983 (2018)
35. Yu, S., Vorobeychik, Y., Alfeld, S.: Adversarial classification on social networks. In: AAMAS. pp. 211–219 (2018)
36. Zhang, M., Hu, L., Shi, C., Wang, X.: Adversarial label-flipping attack and defense for graph neural networks. In: ICDM. pp. 791–800 (2020)
37. Zhang, Y., Humbert, M., Rahman, T., Pang, J., Backes, M.: Tagvisor: A privacy advisor for sharing hashtags. In: WWW (2018)
38. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: SIGKDD. pp. 2847–2856 (2018)
39. Zügner, D., Günnemann, S.: Adversarial attacks on graph neural networks via meta learning. arXiv preprint arXiv:1902.08412 (2019)