Information Obfuscation of Graph Neural Networks

Peiyuan Liao [* 1] **Han Zhao** [* 2] **Keyulu Xu** [* 3]
Tommi Jaakkola [3] **Geoffrey Gordon** [1] **Stefanie Jegelka** [3] **Ruslan Salakhutdinov** [1]

Abstract

While the advent of Graph Neural Networks (GNNs) has greatly improved node and graph representation learning in many applications, the neighborhood aggregation scheme exposes additional vulnerabilities to adversaries seeking to extract node-level information about sensitive attributes. In this paper, we study the problem of protecting sensitive attributes by information obfuscation when learning with graph structured data. We propose a framework to locally filter out pre-determined sensitive attributes via adversarial training with the total variation and the Wasserstein distance. Our method creates a strong defense against inference attacks, while only suffering small loss in task performance. Theoretically, we analyze the effectiveness of our framework against a worst-case adversary, and characterize an inherent trade-off between maximizing predictive accuracy and minimizing information leakage. Experiments across multiple datasets from recommender systems, knowledge graphs and quantum chemistry demonstrate that the proposed approach provides a robust defense across various graph structures and tasks, while producing competitive GNN encoders for downstream tasks.

1. Introduction

Graph neural networks (GNNs) have brought about performance gains in various tasks involving graph-structured data (Scarselli et al., 2009; Xu et al., 2019b). A typical example includes movie recommendation on social networks (Ying et al., 2018). Ideally, the recommender system makes a recommendation not just based on the description of an end user herself, but also those of her close friends in

the social network. By taking the structured information of friendship in social network into consideration, more accurate prediction is often achieved (Xu et al., 2018; Hamilton et al., 2017). However, with better utility comes more vulnerability to potential information leakage. For example, to gain sensitive information about a specific user in the network, malicious adversaries could try to infer sensitive information not just only based on the information of the user of interest, but also information of her friends in the network. Such scenarios are increasingly ubiquitous with the rapid growth of users in common social network platforms, especially in the distributed/federated setting where data are transmitted from local nodes to centralized servers and the goal of malicious adversaries is to reveal users' sensitive data by eavesdropping during the transmission process. Hence, the above problem poses the following challenge:

> *How could we protect sensitive information of users in the network from malicious inference attacks while maintaining the utility of service? Furthermore, can we quantify the potential trade-off between these two goals?*

In this paper, we provide answers to both questions. We propose a simple yet effective algorithm to achieve the first goal through adversarial training of GNNs, a general framework which we term as **G**raph **A**dversaria**L** Networks (GAL). In a nutshell, the proposed algorithm learns node representations in a graph by simultaneously preserving rich information about the target task and filtering information from the representations that is related to the sensitive attribute via a minimax game (Figure 1). While the minimax formulation is not new and has been applied broadly in image generation, domain adaptation and robustness (Goh & Sim, 2010; Goodfellow et al., 2014; Madry et al., 2017; Zhao et al., 2018), we are the first to formulate this problem on graphs for information obfuscation, and to demonstrate that minimax optimization is effective for GNNs in these settings, both theoretically and empirically.

At a high level, for the purpose of information obfuscation of certain sensitive attributes, we propose a method to locally filter out these attributes by learning GNN encoders that could confuse a strong malicious adversary. In particular, the learned GNN encoders will generate node representa-

---

[*]Equal contribution [1]Carnegie Mellon University [2]University of Illinois at Urbana-Champaign [3]Massachusetts Institute of Technology (MIT). Correspondence to: Peiyuan Liao <peiyuanl@andrew.cmu.edu>, Han Zhao <hanzhao@illinois.edu>, Keyulu Xu <keyulu@mit.edu>.

(a) Information Leakage in Client-Server Threat Model
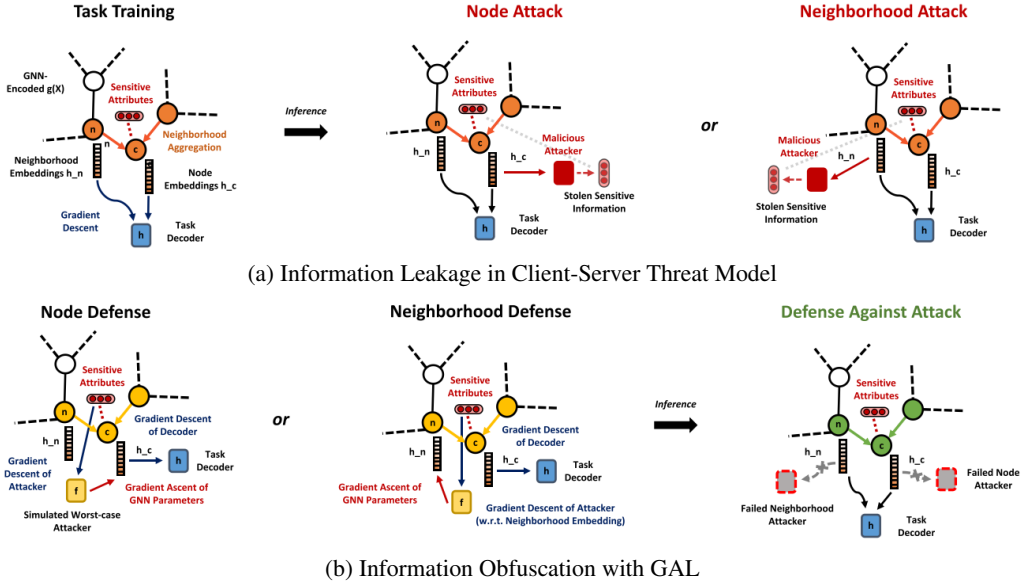

(b) Information Obfuscation with GAL

Figure 1: **Graph Adversarial Networks (GAL).** Above represents a common challenge of inference attack present in graph learning algorithms (a) and our proposed solution (b). For the figure above, starting from left to right, we present a way for adversaries to obtain sensitive information from graph neural networks: after training via neighborhood aggregation (1), by running neural networks on produced embedding of nodes (2) and their neighbors (3) in graphs, they are able to recover sensitive information about them. GAL, on the other hand, defends node and neighborhood inference attacks via a minimax game between the task decoder (blue) and a simulated worst-case adversary (yellow) on both the embedding (descent) and the attributes (ascent). Malicious adversaries will have difficulties extracting sensitive attributes at inference time from GNN embeddings trained with our framework.

tions from which even a worst-case adversary cannot reveal the sensitive attributes. Technically, we use the total variation and the dual formulation of the Wasserstein distance to form a minimax game between the GNN encoder and the adversary, and train it until convergence. We provide theoretical guarantees for our algorithm, and quantify the potential trade-off between GNN predictive accuracy and information leakage. First, we prove a *lower bound* for the inference error over the sensitive attribute that any worst-case adversary has to incur under our algorithm. Second, we quantify how much one has to pay in terms of predictive accuracy for information obfuscation. Specifically, we prove that the loss in terms of predictive accuracy is proportional to how the target task is correlated with the sensitive attribute in input node features.

Empirically, we corroborate our theory and the effectiveness of the proposed framework on 6 graph benchmark datasets. We show that our framework can both train a *competitive* GNN encoder and perform *effective information obfuscation*. For instance, our algorithm successfully decreases the AUC of a gender attacker by 10% on the Movielens dataset while only suffering 3% in task performance. Furthermore, our framework is robust against a new set of attacks we term "neighborhood attacks" or "n-hop attacks", where the adversary can infer node-level sensitive attributes from em-

beddings of 1- and n-distant neighbors. We verify that in these new settings, our algorithm remains effective. Finally, our theoretical bounds on the trade-off between accuracy and defense also agree with experimental results. [1]

In summary, we formulate the information obfuscation problem on GNNs. In this new setting, we show that GNNs trained with GAL achieve both good predictive power and information obfuscation, theoretically (Theorem 3) and empirically (Table 3). Our theory also quantifies the trade-off between accuracy and information leakage (Theorem 1), and is supported by experiments (Figure 6).

2. Preliminaries

We begin by introducing our notation. Let $\mathcal{D}$ be a distribution over a sample space $\mathcal{X}$. For an event $E \subseteq \mathcal{X}$, we use $\mathcal{D}(E)$ to denote the probability of E under $\mathcal{D}$. Given a feature transformation function $g : \mathcal{X} \to \mathcal{Z}$ that maps instances from the input space $\mathcal{X}$ to feature space $\mathcal{Z}$, we define $g_\sharp \mathcal{D} := \mathcal{D} \circ g^{-1}$ to be the pushforward of $\mathcal{D}$ under g, i.e., for any event $E' \subseteq \mathcal{Z}$, $g_\sharp \mathcal{D}(E') := \mathcal{D}(\{x \in \mathcal{X} \mid g(x) \in E'\})$. Throughout the paper we assume that the feature space is bounded, i.e., $\sup_{z \in \mathcal{Z}} \|z\| \le R$. For two distributions $\mathcal{D}$ and $\mathcal{D}'$ over the same sample space

---

[1]Code at: https://github.com/liaopeiyuan/GAL

Ω, let $d_{\text{TV}}(\mathcal{D}, \mathcal{D}')$ be the total variation (TV) distance between them: $d_{\text{TV}}(\mathcal{D}, \mathcal{D}') := \sup_{E \subseteq \Omega} |\mathcal{D}(E) - \mathcal{D}'(E)|$. The Wasserstein distance between $\mathcal{D}$ and $\mathcal{D}'$ is defined as $W_1(\mathcal{D}, \mathcal{D}') := \sup_{\|f\|_L \le 1} \left| \int_\Omega f d\mathcal{D} - \int_\Omega f d\mathcal{D}' \right|$, where $\| \cdot \|_L$ is the Lipschitz semi-norm of a real-valued function.

Graph Neural Networks. GNNs learn node and graph representations for predictions on nodes, relations, or graphs (Scarselli et al., 2009). The input is a graph $G = (V, E)$. Each node $u \in V$ has a feature vector X_u, and each edge $(u, v) \in E$ has a feature vector $X_{(u,v)}$. GNNs iteratively compute a representation for each node. Initially, the node representations are the node features: $X_u^{(0)} = X_u$. In iteration $k = 1, \dots, K$, a GNN updates the node representations $X_u^{(k)}$ by aggregating the neighboring nodes' representations (Gilmer et al., 2017): $\forall k \in [K], u \in \mathcal{N}(v)$,

$$X_v^{(k)} = \text{AGGREGATE}^{(k)} \left(\left\{ \left(X_u^{(k-1)}, X_v^{(k-1)}, X_{(u,v)} \right) \right\} \right).$$

We can compute a graph representation X_G by pooling the final node representations. Many GNNs, with different aggregation and graph pooling functions, have been proposed (Defferrard et al., 2016; Duvenaud et al., 2015; Kipf & Welling, 2017; Hamilton et al., 2017; Du et al., 2019; Fey, 2019; Xu et al., 2020; 2021; Velickovic et al., 2018; Cai et al., 2020).

Threat Model. While being a powerful paradigm for learning node and graph representations for downstream tasks, GNNs also expose huge vulnerability to adversaries whose goal is to infer sensitive attributes from individual node representations. In light of this, throughout the paper we assume a black-box attack setting, where the adversary does not have direct access to the GNN encoder or knowledge of its parameters. Instead, it can obtain the node representations produced by the GNN encoder, with the goal to reconstruct a sensitive attribute A_v of node v by looking at its node representation $X_v^{(K)}$. Here, $A_v \in \{0, 1\}$[2] could be the user's age, gender, or income etc. The above setting is ubiquitous in *server-client* paradigm where machine learning is provided as a service (Ribeiro et al., 2015). For example, in the distributed setting, when users' data is transmitted from local nodes to a centralized server, a malicious adversary could try to reveal users' sensitive attributes by eavesdropping the transmitted data. We emphasize that simply removing the sensitive attribute A_v from the embedding X_v is not sufficient, because A_v may be inferred from some different but correlated features.

Information Obfuscation. We study the problem above from an information-theoretic perspective. Let $\mathcal{F}_A := \{f : \mathbb{R}^{d_K} \to \{0, 1\}\}$ denote the set of adversaries who have access to the node representations as input, and then output

a guess of the sensitive attribute. Let $\mathcal{D}$ be a joint distribution over the node input X, the sensitive attribute A, as well as the target variable Y. Let $\mathcal{D}_a$ be the conditional distribution of $A = a$. We define the *advantage* (Goldwasser & Bellare, 1996) of the adversary as

$$\text{Adv}_\mathcal{D}(\mathcal{F}_A) := \sup_{f \in \mathcal{F}_A} \left| \Pr_{\mathcal{D}_1}(f(Z) = 1) - \Pr_{\mathcal{D}_0}(f(Z) = 1) \right|, \quad (1)$$

where Z is the random variable that denotes *node representations* after applying the GNN encoder to input X. Here, f represents a particular adversary and the supresum in (1) corresponds to the *worst-case adversary* from a class $\mathcal{F}_A$. If $\text{Adv}_\mathcal{D}(\mathcal{F}_A) = 1$, then there exists an adversary who can almost surely guess the sensitive attribute A by looking at the GNN representations Z. Hence, our goal is to design an algorithm that learns GNN representations Z such that $\text{Adv}_\mathcal{D}(\mathcal{F}_A)$ is small, which implies successful defense against adversaries. Throughout the paper, we assume that $\mathcal{F}_A$ is symmetric, i.e., if $f \in \mathcal{F}_A$, then $1 - f \in \mathcal{F}_A$ as well.

3. Information Obfuscation via Adversarial Training

In this section, we first relate the aforementioned *advantage* of the adversaries to a quantity that measures the *ability of adversaries* in predicting a sensitive attribute A by looking at the node representations $Z = X^{(K)}$. Inspired by this relationship, we proceed to introduce a minimax game between the GNN encoder and the worst-case adversary. We then extend this approach by considering adversaries that use score functions induced by the Wasserstein distance. To conclude this section, we analyze the trade-off between accuracy and information leakage, and provide theoretical guarantees for our algorithm.

Given a symmetric set of adversaries $\mathcal{F}_A$, we observe that $1 - \text{Adv}_\mathcal{D}(\mathcal{F}_A)$ corresponds to the sum of best Type-I and Type-II *inference error* any adversary from $\mathcal{F}_A$ could attain:

$$1 - \text{Adv}_\mathcal{D}(\mathcal{F}_A) = \inf_{f \in \mathcal{F}_A} \left(\Pr_{\mathcal{D}_0}(f(Z) = 1) + \Pr_{\mathcal{D}_1}(f(Z) = 0) \right).$$

Hence, to minimize $\text{Adv}_\mathcal{D}(\mathcal{F}_A)$, a natural strategy is to learn parameters of the GNN encoder so that it filters out the sensitive information in the node representation while still tries to preserve relevant information w.r.t. the target task of inferring Y. In more detail, let $Z = X^{(K)} = g(X)$ be the node representations given by a GNN encoder. We then propose the following unconstrained optimization problem with a trade-off parameter $\lambda > 0$ for information obfuscation:

$$\min_{h, Z = g(X)} \max_{f \in \mathcal{F}_A} \quad \varepsilon_Y(h(Z)) - \lambda \cdot \varepsilon_A(f(Z)) \quad (2)$$

Here we use $\varepsilon_Y(\cdot)$ and $\varepsilon_A(\cdot)$ to denote the cross-entropy error on predicting target task Y and sensitive attribute A

---

[2] We assume binary attributes for ease of exposition. Extension to categorical attributes is straightforward.

respectively. In (2), h is a parametrized network over the node representations $X^{(K)}$ for downstream task, and f is the adversary. Note that the above formulation is different from the poisoning attacks on graphs literature (Zügner & Günnemann, 2019), where their goal is to learn a robust classifier under graph perturbation. It is also worth pointing out that the optimization formulation in (2) admits an interesting game-theoretic interpretation, where two agents f and g play a game whose score is defined by the objective function in (2). Intuitively, f could be understood as a simulated adversary seeking to minimize the sum of Type-I and Type-II inference error while the encoder g plays against f by learning features to remove information about the sensitive attribute A. Clearly, the hyperparameter λ controls the trade-off between accuracy and information leakage. On one hand, if $\lambda \to 0$, we barely care about the defense of A and devote all the focus to minimize the predictive error. On the other extreme, if $\lambda \to \infty$, we are only interested in defending against the potential attacks.

Wasserstein Variant In practice, one notable drawback of the optimization formulation in (2) is that the training of the adversary is unstable (cf. Fig. 4), due to the fact that the TV distance is not continuous w.r.t. the model parameters of the encoder (Arjovsky et al., 2017). Inspired by the Wasserstein GAN (Arjovsky et al., 2017), we also propose a method for information obfuscation on GNNs with the Wasserstein distance, using its dual formulation.

$$\min_{h, Z=g(X)} \max_{\|f\|_L \leq 1} \varepsilon_Y(h(Z)) + \lambda \left| \int f(Z)\, d\mathcal{D}_0 - \int f(Z)\, d\mathcal{D}_1 \right|$$

The connection between (2) and the above Wasserstein formulation lies in choice of the adversaries, as can be observed from the comparison between $\mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A)$ (advantage) and $W_1(\cdot, \cdot)$ (Wasserstein distance). In the latter case, the adversaries are constrained to be Lipschitz continuous. In both cases, we present the pseudocode in Fig. 2 and Alg. 1, which also includes extensions to *neighborhood* and *n-hop* attacks (cf. Section 4.3).

3.1. Trade-off between Predictive Accuracy and Information Obfuscation

The first term in the objective function of (2) acts as an incentive to encourage GNN encoders to preserve task-related information. But will this incentive lead to the leakage of the sensitive attribute A? As an extreme case if the target variable Y and the sensitive attribute A are perfectly correlated, then it should be clear that there is a trade-off in achieving accuracy and preventing information leakage. Next, we provide an analysis to quantify this inherent trade-off. Theorem 1 characterizes a trade-off between the cross-entropy error of task predictor and the advantage of the adversaries:

Theorem 1. Let Z be the node representations produced by a GNN g and $\mathcal{F}_A$ be the set of all binary predictors. Define

```
1  z = encoder(graph)
2  z_rev = grad_reverse(z)
3  task_pred, adv_pred = decoder(z),
       attacker(z_rev)
4  loss_task = task_loss_f(task_pred,
       task_gt)
5  loss_adv = adv_loss_f(adv_pred, adv_gt)
6  if t % n == 0
7    optimizer_adv.zero_grad()
8    loss_adv.backward()
9    optimizer_adv.step()
10 else:
11   optimizer_task.zero_grad()
12   loss_task.backward()
13   optimizer_task.step()
```

Figure 2: Pseudocode for the core implementation of GAL (TV distance, node attack). Complete version is in Alg. 1.

$\delta_{Y|A} := |\mathrm{Pr}_{\mathcal{D}_0}(Y=1) - \mathrm{Pr}_{\mathcal{D}_1}(Y=1)|$. Then for a classifier h such that $\|h\|_L \leq C$,

$$\begin{aligned}
& \varepsilon_{Y|A=0}(h \circ g) + \varepsilon_{Y|A=1}(h \circ g) \\
& \geq \delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) \\
& \geq \delta_{Y|A} - 2RC \cdot \mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A). \qquad (3)
\end{aligned}$$

Remark. Recall that R is a bound of the radius of the feature space, i.e., $\sup_{z \in \mathcal{Z}} \|z\| \leq R$. First, $\varepsilon_{Y|A=a}(h \circ g)$ denotes the conditional cross-entropy error of predicting Y given $A = a \in \{0, 1\}$. Hence the above theorem says that, up to a certain threshold given by $\delta_{Y|A}$ (which is a task-specific constant), any target predictor based on the features given by GNN g has to incur a large error on at least one of the sensitive groups. Specifically, the smaller the adversarial advantage $\mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A)$ or the Wasserstein distance $W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1)$, the larger the error lower bound. The lower bound in Theorem 1 is *algorithm-independent*, and considers the strongest possible adversary, hence it reflects an *inherent* trade-off between task utility and information obfuscation. Moreover, Theorem 1 does not depend on the marginal distribution of the sensitive attribute A. Instead, all the terms in our result only depend on the conditional distributions given $A = 0$ and $A = 1$. As a simple corollary, the overall error also admits a lower bound:

Corollary 2. Assume the conditions in Theorem 1 hold. Let $\alpha := \mathrm{Pr}_{\mathcal{D}}(A = 0)$, then

$$\begin{aligned}
\varepsilon_Y(h \circ g) & \geq \min\{\alpha, 1-\alpha\}\big(\delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1)\big) \\
& \geq \min\{\alpha, 1-\alpha\}\big(\delta_{Y|A} - 2RC \cdot \mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A)\big).
\end{aligned}$$

Our lower bounds in Theorem 1 and Corollary 2 capture the price we have to pay for obfuscation.

3.2. Guarantees Against Information Leakage

Next, we provide guarantees for information obfuscation using (2). The analysis on the optimization trajectory of a

general non-convex-concave game (2) is still an active area of research (Daskalakis & Panageas, 2018; Nouiehed et al., 2019) and hence beyond the scope of this paper. Therefore, we assume that we have access to the minimax stationary point solution of (2), and focus on understanding how the solution of (2) affects the effectiveness of our defense.

In what follows we analyze the true error that a worst-case adversary has to incur in the limit, when both the task classifier and the adversary have unlimited capacity, i.e., they can be any randomized functions from $\mathcal{Z}$ to $\{0, 1\}$. To this end, we also use the population loss rather than the empirical loss in our objective function. Under such assumptions, given any node embedding Z from a GNN g, the worst-case adversary is the *conditional mean*: $\min_{f \in \mathcal{F}_A} \varepsilon_A(f \circ g) = H(A \mid Z)$, $\arg\min_{f \in \mathcal{F}_A} \varepsilon_A(f \circ g) = \Pr(A = 1 \mid Z)$. It follows from a symmetric argument that $\min_{h \in \mathcal{H}} \varepsilon_Y(h \circ g) = H(Y \mid Z)$. Hence we can simplify the objective (2) to the following form where the only variable is the embedding Z:

$$\min_{Z = g(X)} \quad H(Y \mid Z) - \lambda \cdot H(A \mid Z) \qquad (4)$$

We can now analyze the error that has to be incurred by the worst-case adversary:

Theorem 3. Let Z^* be the optimal GNN node embedding of (4). Define $\alpha := \Pr_{\mathcal{D}}(A = 0)$, $H^* := H(A \mid Z^*)$ and $W_1^* := W_1(Z^* \mid A = 0, Z^* \mid A = 1)$. Then 1). For any adversary $f : \mathcal{Z} \to \{0, 1\}$, $\Pr(f(Z) \neq A) \geq H^*/2 \lg(6/H^*)$, 2). For any Lipschitz adversary f such that $\|f\|_L \leq C$, $\Pr(f(Z) \neq A) \geq \min\{\alpha, 1 - \alpha\}(1 - CW_1^*)$.

Theorem 3 shows that whenever the conditional entropy $H^* = H(A \mid Z^*)$ is large or the Wasserstein distance W_1^* is small, the inference error incurred by any (randomized) adversary has to be large. It is worth pointing out that when $W_1^* = 0$, the second inference error lower bound reduces to $\min\{\alpha, 1 - \alpha\}$, which is attained by an adversary that uses constant prediction of the sensitive attribute A, i.e., this adversary always guesses the majority value of A. Hence, Theorem 3 justifies the use of GAL for information obfuscation. As a final note, recall that the representations Z appearing in the bounds above depend on the *graph structure*, and the inference error in Theorem 3 is over the representations Z. Together, this suggests that the defense could potentially be applied against *neighborhood attacks*, which we provide in-depth empirical validation in Section 4.3.

4. Experiments

In this section, we demonstrate the effectiveness of GAL for information obfuscation on graphs. Specifically, we address the following three questions:

➜ 4.1: Is GAL effective across different tasks, distances and GNN encoder architectures? How do the TV and Wasserstein variants compare during training?

Algorithm 1 Full description of GAL, TV distance and Wasserstein distance. The node pairing policy Q decides how the predicted node-level sensitive attributes A_v matches against the targets A_w for all nodes in the graph: in node-level obfuscation, it is simply the identity transformation; for neighborhood and N-Hop attack, pairing is done with the proposed Monte-Carlo probabilistic algorithm on G. Note that for the expository analysis above , $L_{\text{adversary}}$ is equivalent to ε_A.

Input: $G = (V, E)$: input graph; Y: node-level target; A: node-level sensitive attributes; g: GNN encoder; h: task decoder; f: worst-case adversary; L_{task}: task loss; $L_{\text{adversary}}$: adversary loss; Q: node-pairing policy
Input: n: training iterations; α: learning rate; m: batch size; n_a: adversary per task; λ: trade-off parameter
Input: $\theta_g, \theta_h, \theta_f$: corresponding parameters for network
Input: (Required by Wasserstein) c: clipping parameter; C_A: classes in A
$t = 0$
repeat
  $z = g(V, E)$ {via AGGREGATE}
  $z_{\text{rev}} = \text{GradReverse}(z)$ (Ganin et al., 2016)
  $Y' = h(z)$
  $A' = f(z_{\text{rev}})$
  if $t \bmod n_a = 0$ **then**
    $B = (A_p, A'_p) = Q(A, A')$
    (TV): $l = L_{\text{adversary}}(A_p, A'_p)$,
    (Wasserstein): $\bar{p} = \text{mean}(\{(A_p^{'(a)}) \odot 1_{A_p = a} : a \in C_A)$
    $l = |\max_{a \in C_A}(\bar{p}_a) - \min_{a \in C_A}(\bar{p}_a)|$
    $g_g, g_f = \nabla_g l, \nabla_f l$
    $\theta_g, \theta_f = \theta_g + \lambda \cdot \text{SGD}(\theta_g, g_g), \theta_f + \lambda \cdot \text{SGD}(\theta_f, g_f)$
    (Wasserstein): $\theta_f = \text{clip}(\theta_f, -c, c)$
  else
    $g_g, g_h = \nabla_g L_{\text{task}}(Y, Y'), \nabla_h L_{\text{task}}(Y, Y')$
    $\theta_g, \theta_h = \theta_g + \alpha \cdot \text{SGD}(\theta_g, g_g), \theta_h + \alpha \cdot \text{SGD}(\theta_h, g_h)$
  end if
  $t = t + 1$
until $t >= n$

➜ 4.2: What is the landscape of the tradeoff with respect to the hyperparameter λ?

➜ 4.3: Can GAL also defend neighborhood and n-hop attacks?

We empirically confirm all three questions. To stress test the robustness of GAL, we consider a variety of tasks and GNN encoder architectures. Specifically, we experiment on 5 link-prediction benchmarks (Movielens-1M, FB15k-237, WN18RR, CiteSeer, Pubmed) and 1 graph regression benchmark (QM9), which covers both obfuscation of *single* and *multiple* attributes. More importantly, our goal is not to challenge state-of-the-art training schemes, but to observe the

Table 1: **Performance of prediction and obfuscation on Movielens-1M**. We first perform link prediction as the main task, then probe with adversaries performing node classification tasks. The statistics are reported as {adversary performance, F1/AUC}/{task performance, RMSE}. Best-performing methods and best obfuscation results are highlighted in bold in the first table. Since Bose & Hamilton (2019a)'s method does not allow tradeoff tuning, incompatible fields are left unfilled.

Encoder	λ	Gender-F1/Task-RMSE	Gender-AUC/Task-RMSE	Age-F1/Task-RMSE	Occupation-F1/Task-RMSE
ChebNet-TV	0.5	0.617 / 0.881	0.642 / 0.881	0.141 / 0.875	0.044 / 0.868
ChebNet-W	0.5	**0.593** / 0.872	**0.605** / 0.872	**0.137** / 0.888	**0.033** / 0.908
GraphSAGE-TV	0.5	0.679 / 0.866	0.680 / 0.866	0.236 / 0.869	0.055 / 0.871
GraphSAGE-W	0.5	0.691 / 0.893	0.715 / 0.893	0.226 / 0.901	0.050 / 0.916
Bose and Hamilton		0.667 / 0.874	0.678 / 0.874	0.188 / 0.874	0.051 / 0.874

ChebNet-TV	0	0.692 / 0.852	0.707 / 0.852	0.288 / 0.852	0.078 / 0.851	ChebNet-TV	4	0.505 / 1.280	0.532 / 1.280	0.123 / 1.301	0.016 / 1.241
ChebNet-W	0	0.693 / 0.852	0.707 / 0.852	0.286 / 0.852	0.077 / 0.851	ChebNet-W	4	0.485 / 1.258	0.526 / 1.258	0.104 / 1.296	0.025 / 1.353
GraphSAGE-TV	0	0.728 / 0.849	0.735 / 0.849	0.293 / 0.849	0.080 / 0.851	GraphSAGE-TV	4	0.675 / 0.900	0.683 / 0.900	0.200 / 0.904	0.050 / 0.898
GraphSAGE-W	0	0.724 / 0.849	0.734 / 0.849	0.293 / 0.849	0.081 / 0.851	GraphSAGE-W	4	0.471 / 0.970	0.516 / 0.970	0.074 / 1.080	0.010 / 1.117

Table 2: **Summary of benchmark dataset statistics**, including number of nodes $|V|$, number of nodes with sensitive attributes $|S|$, number of edges $|E|$, node-level non-sensitive features d, target task and adversarial task, and whether the experiment concerns with multi-set obfuscation.

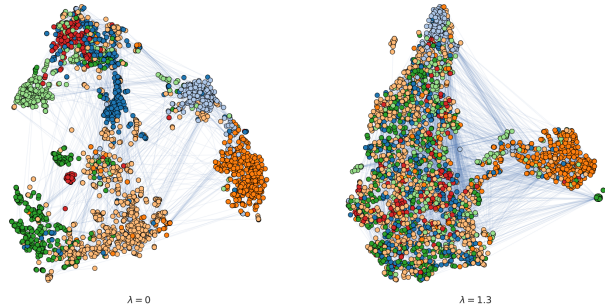| DATASET	$|V|$	$|S|$	$|E|$	d	Multi?	METRICS	ADVERSARY
CITESEER	3,327	3,327	4,552	3,703	✗	AUC	Macro-F1
PUBMED	19,717	19,717	44,324	500	✗	AUC	Macro-F1
QM9	2,383,055	2,383,055	2,461,144	13	✗	MAE	MAE
ML-1M	9,940	6,040	1,000,209	1 (id)	✗	RMSE	Macro-F1/AUC
FB15k-237	14,940	14,940	168,618	1 (id)	✓	MRR	Macro-F1
WN18RR	40,943	40,943	173,670	1 (id)	✗	MRR	Macro-F1

Figure 3: **GAL effectively protects sensitive information**. Both panels show t-SNE of feature representations under different trade-off parameters λ (0 vs. 1.3). Node colors represent node classes.

effect in reducing adversaries' accuracies while maintaining good performance of the downstream task. Our experiments can be classified into three major categories.

Robustness. We run GAL on all six datasets under a variety of GNN architectures, random seeds, distances and trade-off parameter λ. We report average performance over five runs. We perform ablation study on widely used GNN architectures (Velickovic et al., 2018; Kipf & Welling, 2017; Vashishth et al., 2019; Defferrard et al., 2016; Gilmer et al., 2017), and select the best-performing GNNs in the task: CompGCN is specifically designed for knowledge-graph-related applications.

Trade-off. We compare performance under a wide range of λ on Movielens-1M. We show that GAL defends the inference attacks to a great extent while only suffering minor losses to downstream task performance. We also empirically compare the training trajectories of the TV and Wasserstein variants, and confirm that the Wasserstein variant often leads to more stable training.

Neighborhood Attacks. In this setting, an adversary also has access to the embeddings of *neighbors* of a node, e.g. the adversary can infer sensitive attribute A_v from $X_w^{(K)}$ (instead of $X_v^{(K)}$) such that there is a path between v and w. Since GNN's neighborhood-aggregation paradigm may introduce such information leakage, adversaries shall achieve nontrivial performance. We further generalize this to an n-hop scenario.

In all experiments, the adversaries only have access to

the training set labels along with embeddings from the GNN, and the performance is measured on the held-out test set. A summary of the datasets, including graph attributes, task, and adversary metrics, is in Table 2. Detailed experimental setups may be found in Appendix B. Overall, our results have successfully addressed all three questions, demonstrating that our framework is attractive for node- and neighborhood-level attribute obfuscation across downstream tasks, GNN architectures, and trade-off parameter λ.
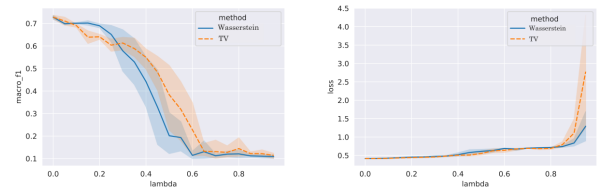
(a) Adversary: Macro F-1   (b) Task: Binary cross-entropy

Figure 4: **Performance of Wasserstein vs. TV for node-level information obfuscation** on Cora dataset under different λ, with 95% confidence interval over five runs. Wasserstein offers a more stable task performance while producing a better obfuscation compared to TV.
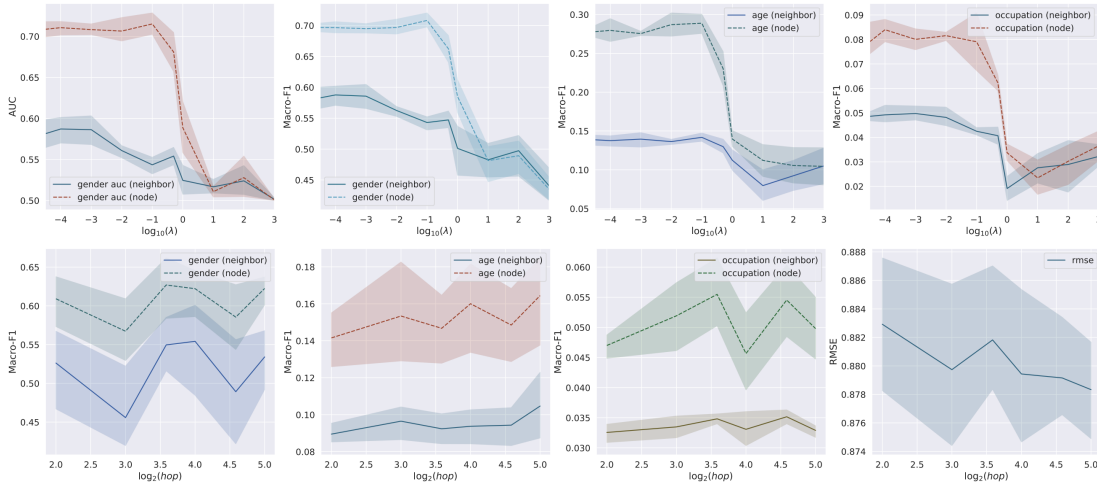
Figure 5: **Performance of neighborhood-level (top) and n-hop (bottom) attribute obfuscation** on a 3-layer/2-layer ChebNet with Movielens-1M dataset under different λ/"hop" respectively. Band represents 95% confidence interval over five runs. For n-hop experiments, λ is fixed to be 0.8, and distance is Total Variation.
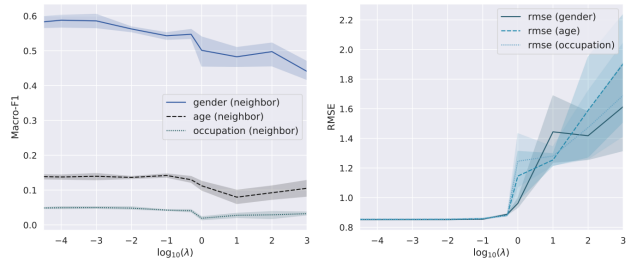

Figure 6: **Performance of node-level attribute obfuscation** on Movielens-1M dataset under different λ, with 95% confidence interval over five runs.

4.1. Robust Node-level Information Obfuscation

We first demonstrate that our framework is effective across a wide variety of GNN architectures, tasks and training methods. Quantitatively, we report task performances T_i as well as adversary performances $A_i, \forall i \in [3]$ both specific to respective datasets in Table 1 and Table 3, where we use the subscript i to denote the i-th experimental setting. Note that fixed-embedding (Bose & Hamilton, 2019b) results are only reported for Movielens-1M dataset because on other datasets, the experimental setups are not suitable for comparison. Across all datasets, we witness a significant drop in adversary's performance with a rather minor decrease in the downstream task performance.

Qualitatively, the effect of adversarial training is also apparent: in Figure 3, we visualized the t-SNE (van der Maaten & Hinton, 2008)-transformed representations of graph embeddings under different regularization strengths on the Cora dataset. We observe that under a high regularization strength, node-level attributes are better mixed in the sense that nodes belonging to different classes are harder to separate from each other on a graph-structural level.

4.2. Fine-grained Trade-off Tuning with λ

We show that the hyper-parameter λ is indeed a good trade-off hyper-parameter that well balances downstream task performance and obfuscation effectiveness. We perform experiments with ChebNet and the Movielens-1M dataset, and test λ values ranging from 0 to 10^3. Our results in Figure 6 show that as λ increases, it is indeed more difficult for the adversary to extract sensitive information from GNN embeddings; the downstream task performance decreases steadily but with a small rate. When λ is larger than 10, training destabilizes further due to inherent difficulties in optimization, which is demonstrated by the higher variance in task performance. However, this trade-off is less visible for larger graphs like FB15k-237 and WN18RR, where the effects of lambda tends to be less monotonic. The detailed results are in Table 3 and Figure 4.

4.3. Neighborhood Attack and n-Hop Genrealization

Finally, we consider the scenario where the adversary seeks to extract node-level information from embeddings of *neighbors*. This is, in particular, a serious concern in GNNs where the message passing layers incorporate neighborhood-level information and hence the information leakage. Our results in Figure 5 verify that, without any form of obfuscation ($\lambda = 0$), the adversary indeed achieves nontrivial performance by accessing the embeddings of neighbors on the Movielens-1M dataset with a simple 2-layer ChebNet. We further run comprehensive experiments where the obfuscation targets neighborhood-level embeddings, and we observe the degradation of the adversary's performance (both on the node- and neighbor-level embeddings) as λ increases. The results demonstrate that under low values of λ, the neighborhood-level obfuscation has more effects on neighborhood adversaries than node adversaries, suggesting that

Table 3: **Performance of predictions and obfuscation on benchmark datasets**. λ_i, T_i, A_i represents the tradeoff parameter, task performance and adversary performance in the i-th experiment, respectively. Best-performing GNNs and best obfuscation results are highlighted in bold. The "—" mark stands for "same as above."

	GNN Encoder	Method	Adversary	λ_1	λ_2	λ_3	T_1	T_2	T_3	A_1	A_2	A_3
PUBMED	GCN	Total Variation	Doc. Class	0.5	0.85	0.95	**0.959**	0.930	0.818	0.800	0.798	0.762
		Wasserstein	Doc. Class	—	—	—	0.959	0.895	0.733	0.798	0.780	0.596
	GAT	Total Variation	Doc. Class	—	—	—	0.924	0.873	0.810	0.782	0.794	0.730
		Wasserstein	Doc. Class	—	—	—	0.928	0.846	0.785	0.778	0.766	0.741
	ChebNet	Total Variation	Doc. Class	—	—	—	0.933	0.872	0.757	0.798	0.788	0.747
		Wasserstein	Doc. Class	—	—	—	0.919	0.727	0.739	0.780	0.641	**0.568**
CITESEER	GCN	Total Variation	Doc. Class	0.75	1	1.5	0.867	0.694	0.500	0.785	0.731	0.652
		Wasserstein	Doc. Class	—	—	—	0.814	0.623	0.504	0.644	0.608	0.407
	GAT	Total Variation	Doc. Class	—	—	—	**0.896**	0.553	0.505	0.789	0.273	0.168
		Wasserstein	Doc. Class	—	—	—	0.811	0.519	0.501	0.656	0.278	**0.125**
	ChebNet	Total Variation	Doc. Class	—	—	—	0.776	0.678	0.500	0.735	0.725	0.642
		Wasserstein	Doc. Class	—	—	—	0.732	0.614	0.500	0.661	0.568	0.403
QM9	MPNN	Total Variation	Polarizability	0	0.05	0.5	**0.121**	0.132	0.641	1.054	1.359	**3.100**
WN18RR	CompGCN	Total Variation	Word Sense	0	1.0	1.5	**0.462**	0.437	0.403	0.208	**0.131**	0.187
		Total Variation	POS tag	—	—	—	—	0.430	0.395	0.822	**0.607**	0.705
FB15K-237	CompGCN	Total Variation	Ent. Attr.	0	1.0	1.5	**0.351**	0.320	0.319	0.682	0.641	**0.630**

node-level embeddings are less protected. However, as λ continues to increase, the degradation in the performance of node adversaries is more visible. An extension of this setup is to consider neighborhood-level attacks as "single-hop", where we can naturally define "n-hop" as adversaries having accesses to only embeddings of nodes that are n-distant away from the target node.

Since finding n-distant neighbors for arbitrary nodes in a large graph on-the-fly is computationally inefficient during training (in the sense that the complexity bound involves $|E|$ and $|V|$), we propose a Monte-Carlo algorithm that probabilistically finds such neighbor in $O(n^2)$ time, the details of which can be found in appendix. We report results under different "hops" with the same encoder-λ setting on Movielens-1M, shown in Figure 5. In general, we observe that as "hop" increases, the retrieved embedding contains less information about the target node. Therefore, adversarial training's effect will have less effect on the degradation of the target task, which is demonstrated by the steady decrease of RMSE. The fluctuations in the trend of node- and neighborhood-level adversaries are due to the probabilistic nature of the Monte-Carlo algorithm used to sample neighbors, where it may end up finding a much closer neighbor than intended, destabilizing the training process. This is another trade-off due to limited training time, yet the general trend is still visible, certifying our assumptions.

5. Other Related Work

Adversarial Attack on Graphs. Our problem formulation, i.e., information obfuscation, is *significantly different* from adversarial attacks on graphs (Bojchevski & Günnemann, 2019; Ma et al., 2019; Xu et al., 2019a; Dai et al., 2018;

Chang et al., 2019). Works on adversarial attacks focus on perturbations of a graph, e.g., by adding or removing edges, that maximize the loss of "victim" nodes for GNNs. In contrast, adversaries in our framework do not alter graph structure; instead, they seek to extract critical information from the GNN node embeddings.

Differential Privacy. A related but *different* notion of privacy is differential privacy (Dwork et al., 2014), which aims at defending the so-called membership inference attack (Shokri et al., 2017; Nasr et al., 2018). Their goal is to design randomized algorithms so that an adversary cannot guess whether an individual appears in the training data by looking at the output of the algorithm. We instead tackle a information obfuscation problem where the adversary's goal is to infer some attribute of a node in a graph, and our goal is to find embedding of the data, so that the adversary cannot accurately infer a sensitive attribute from the embedding.

Information Bottleneck. Our work is also related to but *different from* the information bottleneck method, which seeks to simultaneously compress data while preserving information for target tasks (Tishby et al., 2000; Alemi et al., 2016; Tishby & Zaslavsky, 2015). The information bottleneck framework could be understood as maximizing the following objective: $I(Y; Z) - \beta I(X; Z)$. Specifically, there is no sensitive attribute A in the formulation. In contrast, the minimax framework in this work is about a trade-off problem, and the original input X does not appear in our formulation: $I(Y; Z) - \beta I(A; Z)$. Overall, despite their surface similarity, our adversarial method is significantly different from that of the information bottleneck method in terms of both formulation and goals.

6. Conclusion

In this work, we formulate and address the problem of information obfuscation on graphs with GNNs. Our framework, termed GAL, introduces a minimax game between the desired GNN encoder and the worst-case adversary. GAL creates a strong defense against information leakage in terms of a provable lower bound, while only suffering a marginal loss in task performance. We also show an information-theoretic bound for the inherent trade-off between accuracy and obfuscation. Experiments show that GAL perfectly complements existing algorithms deployed in downstream tasks to address information security concerns on graphs.

Acknowledgements

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Berg, R. v. d., Kipf, T. N., and Welling, M. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

Bojchevski, A. and Günnemann, S. Adversarial attacks on node embeddings via graph poisoning. In *ICML*, 2019.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 2013. to appear.

Bose, A. and Hamilton, W. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pp. 715–724, 2019a.

Bose, A. and Hamilton, W. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pp. 715–724, 2019b.

Cai, T., Luo, S., Xu, K., He, D., Liu, T.-y., and Wang, L. Graphnorm: A principled approach to accelerating graph neural network training. *arXiv preprint arXiv:2009.03294*, 2020.

Calabro, C. *The exponential complexity of satisfiability problems*. PhD thesis, UC San Diego, 2009.

Chang, H., Rong, Y., Xu, T., Huang, W., Zhang, H., Cui, P., Zhu, W., and Huang, J. The general black-box attack method for graph neural networks. *ArXiv*, abs/1908.01297, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial attack on graph structured data. In *ICML*, 2018.

Daskalakis, C. and Panageas, I. The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems*, pp. 9236–9246, 2018.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. pp. 3844–3852, 2016.

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Advances in Neural Information Processing Systems*, pp. 5724–5734, 2019.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. pp. 2224–2232, 2015.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Fey, M. Just jump: Dynamic neighborhood aggregation in graph neural networks. *arXiv preprint arXiv:1904.04849*, 2019.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Gibbs, A. L. and Su, F. E. On choosing and bounding probability metrics. *International statistical review*, 70 (3):419–435, 2002.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1273–1272, 2017.

Goh, J. and Sim, M. Distributionally robust optimization and its tractable approximations. *Operations research*, 58 (4-part-1):902–917, 2010.

Goldwasser, S. and Bellare, M. Lecture notes on cryptography. *Summer course "Cryptography and computer security" at MIT*, 1999:1999, 1996.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. pp. 1025–1035, 2017.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Landrum, G. Rdkit: Open-source cheminformatics. URL http://www.rdkit.org.

Ma, Y., Wang, S., Wu, L., and Tang, J. Attacking graph convolutional networks via rewiring. *ArXiv*, abs/1906.03750, 2019.

Maas, A. L. Rectifier nonlinearities improve neural network acoustic models. 2013.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Moon, C., Jones, P., and Samatova, N. F. Learning entity type embeddings for knowledge graph completion. In *Proceedings of the 2017 ACM on conference on information and knowledge management*, pp. 2215–2218, 2017.

Nasr, M., Shokri, R., and Houmansadr, A. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 634–646, 2018.

Nouiehed, M., Sanjabi, M., Huang, T., Lee, J. D., and Razaviyayn, M. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, pp. 14905–14916, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

Ribeiro, M., Grolinger, K., and Capretz, M. A. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 896–902. IEEE, 2015.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne, 2008.

Vashishth, S., Sanyal, S., Nitin, V., and Talukdar, P. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. 2018.

Vinyals, O., Bengio, S., and Kudlur, M. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. S. Moleculenet: A benchmark for molecular machine learning. 2017.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pp. 5453–5462, 2018.

Xu, K., Chen, H., Liu, S., Chen, P.-Y., Weng, T.-W., Hong, M., and Lin, X. Topology attack and defense for graph neural networks: An optimization perspective. In *IJCAI*, 2019a.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019b.

Xu, K., Li, J., Zhang, M., Du, S. S., ichi Kawarabayashi, K., and Jegelka, S. What can neural networks reason about? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJxbJeHFPS.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-I., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=UH-cmocLJC.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 974–983, 2018.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pp. 8559–8570, 2018.

Zügner, D. and Günnemann, S. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 246–256, 2019.