# Adversarial Label Poisoning Attack on Graph Neural Networks via Label Propagation

Ganlin Liu, Xiaowei Huang, and Xinping Yi

University of Liverpool, Liverpool, United Kingdom
{ganlin.liu, xiaowei.huang, xinping.yi}@liverpool.ac.uk

**Abstract.** Graph neural networks (GNNs) have achieved outstanding performance in semi-supervised learning tasks with partially labeled graph structured data. However, labeling graph data for training is a challenging task, and inaccurate labels may mislead the training process to erroneous GNN models for node classification. In this paper, we consider label poisoning attacks on training data, where the labels of input data are modified by an adversary before training, to understand to what extent the state-of-the-art GNN models are resistant/vulnerable to such attacks. Specifically, we propose a label poisoning attack framework for graph convolutional networks (GCNs), inspired by the equivalence between label propagation and decoupled GCNs that separate message passing from neural networks. Instead of attacking the entire GCN models, we propose to attack solely label propagation for message passing. It turns out that a gradient-based attack on label propagation is effective and efficient towards the misleading of GCN training. More remarkably, such label attack can be topology-agnostic in the sense that the labels to be attacked can be efficiently chosen without knowing graph structures. Extensive experimental results demonstrate the effectiveness of the proposed method against state-of-the-art GCN-like models.

**Keywords:** Label Poisoning Attack, Graph Neural Networks, Label Propagation, Graph Convolutional Network

## 1 Introduction

The past years have witnessed the increasing interest in studying computer vision (CV) and machine learning tasks using graph representation learning, especially graph neural networks (GNNs). Differently from the canonical convolutional neural network (CNN) models, GNNs explore graph data structures for various CV applications, e.g., semantic object parsing [12], skeleton-based action recognition [32], visual tracking [6], video parsing [28], point cloud classification [29], to name just a few. In particular, scene graphs can be constructed from parsing images/videos to form the semantic relationships between pairs of objects, for which GNNs play a role in image classification/recognition/segmentation. In terms of node classification, GCN models have been also used in medical image analysis for disease prediction to classify the health status of the patient or the type of case [19, 31].

Besides the emerging CV applications, GNN models have been demonstrated a powerful tool in a variety of semi-supervised learning tasks, such as node classification [10, 26, 7], graph classification [34], and link prediction [36], in many application scenarios in our social life, such as social networks [5], knowledge graphs [27] and recommendation systems [13, 33]. For example, given graph structured data with node features and partial labels, node classification is to predict the labels of those unlabeled data according to the inter-node relationship modeled as a graph. The state-of-the-art GNN models leverage both feature transformation that maps input features to graph embeddings via neural networks and neighborhood aggregation via a message passing mechanism to predict the labels.

However, in practical applications, labeling graph data is a challenging task; limited or inaccurate labels could lead to erroneous trained models via message passing. It is crucial to understand as to how GNN models are resistant or vulnerable to limited or inaccurate labels and to what extent we could rely on the labeled data in graph representation learning. To this end, we consider label poisoning attack in the *training* phase to reveal the potential vulnerability of GNN training, where an adversary attempts to decrease the node classification accuracy of unlabeled data by modifying the labels of the known ones before training. It is worth noting that adversarial label poisoning attack is different from the intensively studied adversarial attack (e.g. [25, 8]) in the *inference* phase.

Compared with the inference time attack, less attention has been paid to the poisoning attack on GNN models. Of the most relevance is the recent work, called Label-flipping attacK model against GNNs (LafAK), which generates gradient-based attacks based on an approximated closed form of the entire GNN model and continuous surrogates of attack objectives. Although promising from a conceptual point of view, LafAK is restricted to binary classification and relies highly on the overall loss function of the entire GNN models, which involves model parameters of neural networks that are unknown before training although an approximate closed-form solution to a linearized model is replaced.

In this paper, we aim to address the above issues by proposing a novel adversarial label poisoning attack for GNN models via label propagation, which captures the essence of massage passing whilst leaving aside the unknown model parameters of neural networks for feature transformation. Instead of attacking the entire GNN models with intertwined neural feature transformation and neighborhood aggregation, we solely attack the neighborhood aggregation of message passing that can be captured by a label propagation process. The rationale behind this idea is underpinned by recent advances on the equivalence between decoupled graph convolution network (DGCN) and label propagation [4]: training DGCN with separated feature transformation and neighborhood aggregation (e.g., APPNP [11] and SGCN [30]), is equivalent to label propagation to generate pseudo-labels for unlabeled data followed by training neural network with pseudo-labeled data with dynamic weights. It turns out attacking the loss function of the label propagation process alone to locate the label poisoning is effective and efficient in poisoning the entire GNN models.

Specifically, the contributions of our work can be summarized as follows:

- We propose a label poisoning attack framework for GCN-like models with three components: label propagation to generate predictive label, maximum gradient attack method to reduce the accuracy of an equivalent label propagation model, and GNN training with poisoned labels. Remarkably, our proposed label poisoning attack works for multi-class node classification and does not require to access the model parameters of neural networks in GCNs.
- We propose a maximum gradient attack method to poison known labels that maximizes the loss function of equivalent label propagation models. Among different label propagation models, we found that the knowledge of the graph structure is not necessary, but the node features are sufficient to yield an effective attack.
- We conduct an extensive set of experiments with different choices of label propagation models to test the attack performance against various GCN-like models (e.g., GCN [10], GAT [26], GraphSAGE [7], APPNP [11], SGCN [30], PT [4]) for multi-class node classification and demonstrate the effectiveness and efficiency of our proposed method.

## 2 Preliminaries and Related Work

### 2.1 Graph Convolution Network and Its Decoupled Variants

**Graph Convolution Network (GCN).** Roughly speaking, GCN is a feature extractor for graph structured data modeled by a graph topology $\mathcal{G}$ with $n$ nodes specified by its adjacency matrix $A \in \{0,1\}^{n\times n}$ and the input features $X \in \mathbb{R}^{n\times f}$ with each row being node attribute of one node and $f$ being the number of features. It produces node embeddings from input features through a message passing mechanism over $\mathcal{G}$ with all node equipped with neural networks for feature transformation. The message passing performs iterative message aggregation/propagation from/to neighboring nodes as if stacking a number of layers. In each iteration/layer, each node employs a multilayer perceptron (MLP) to transform node features into a new embedding and propagates it to its neighbors, followed by feature aggregation from neighboring nodes to update node features for the next iteration/layer [10]. A typical GCN layer for node feature transformation can be written as

$$H^{(k+1)} = \sigma(\hat{A}H^{(k)}W^{(k)}) \tag{1}$$

where $\hat{A}$ is the normalized adjacency matrix, $H^{(k)}$ is the updated node feature representations after neighborhood message aggregation at $k$-th layer with $H^{(0)} = X$ being the input feature matrix, $\sigma(\cdot)$ denotes a nonlinear activation function, such as ReLU, and $W^{(k)}$ is the trainable weight matrix of the MLP at the $k$-th layer. In the original GCN model [10], $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is adopted, where $\tilde{A} = A + I$ with $I$ being the identity matrix, and $\tilde{D} = \mathrm{diag}(d_1, ..., d_n)$ is the degree matrix with $d_i = \sum_j A_{ij}$ being the degree of the node $i$.

The above vanilla GCN models (e.g., [10, 7]) combine intertwined feature transformation and neighborhood aggregation for node representation learning.

The extracted node feature representations can be used in many graph learning tasks such as node classification [11, 7], graph classification [20], link prediction [36] and graph embedding [22, 1].

**Decoupled Graph Convolutional Network (DGCN).** Differently from the vanilla GCN models with coupled feature transformation and neighborhood aggregation, some recent studies have found that such coupled designs are unnecessary and have proposed to separate these two operations. These decoupled GCN (DGCN) models (e.g., [4]) can be summarized as

$$\hat{Y} \;=\; \text{softmax}(\bar{A} f_\theta(X)) \tag{2}$$

where $f_\theta(\cdot)$ is usually a neural network with parameters $\theta$ to transform input features $X$ to certain representations, and $\bar{A}$ is a propagation matrix determined by the propagation strategies that will be specified later.

In what follows, we briefly describe two existing DGCN models: Approximate Personalized Propagation of Neural Predictions (APPNP) [11] and Simplifying Graph Convolutional Network (SGCN) [30].

**APPNP** [11] separates the neural network from the propagation scheme. It uses Personalized PageRank [18] as the propagation strategy in order to leverage a larger neighborhood information than GCN, and can be combined with any state-of-the-art neural network. Thus, the propagation matrix $\bar{A}$ in (2) can be specified by

$$\bar{A} = (1-\alpha)^K \hat{A}^K + \alpha \sum_{k=0}^{K-1} (1-\alpha)^k \hat{A}^k \tag{3}$$

where $\alpha \in (0, 1]$ is the teleport probability and $K$ is the number of layers.

**SGCN** [30] aims to transform the nonlinear GCN into a simple linear model. By successively removing the nonlinear transition functions and collapsing weight matrices between GCN layers, the additional complexity of GCNs can be reduced. The resulting simplified linear model can be replaced by (2) with

$$\bar{A} = \hat{A}^K \text{ and } f_\theta(X) = X\Theta \tag{4}$$

where $K$ is the number of SGCN layers, and $\Theta$ is a reparameterized weight matrix with non-linearity removed.

### 2.2 Label Propagation and Its Role in GCN

**Label propagation (LP)** algorithm [38] is a commonly used semi-supervised learning method in machine learning, propagating known labels from labeled nodes to unlabeled ones. An implicit assumption of LP is that similar data should have the same label. For each node in the network, at the initial stage, the label propagation algorithm initializes a unique label for each node. Each iteration will change its own label according to the label of the node connected to it. The way of change is to select the community label with the most labels among the nodes connected to it as its own community label. As community labels spread, eventually, tightly connected nodes will have common labels.

**Propagation then Training (PT)** [4] is a variant of label propagation algorithm, including two steps: (1) Label propagation, where the known labels are propagated along the edges of graph to generate pseudo-labels for unlabeled nodes; (2) Neural network training, where feature transformation is done by training a neural network classifier on the data with known and pseudo-labels. The loss function of PT can be written as

$$L(\theta) = \ell(f_\theta(X), \bar{A}Y) \tag{5}$$

where $\ell(\cdot)$ is the loss function between $f_\theta(X)$ and the soft labels $Y_{soft} = \bar{A}Y$.
**Equivalence of Decoupled GCN and PT.** Hande et al. (2021) [4] discusses decoupled GCN of semi-supervised node classification from a new perspective. It is proved that the decoupled GCN is essentially the same as two-step label propagation via in-depth theoretical analysis, and the effectiveness of decoupled GCN is also revealed.

### 2.3 Adversarial Poisoning Attacks on Graphs

Adversarial attack lies in the intersection of machine learning and cyber-security. It has been evidenced that deep learning models often suffer from adversarial attacks with degraded accuracy performance. Roughly speaking, adversarial attacks can be categorized into evasion and poisoning attacks according to the stage when attacks occur. While evasion attacks happen in the testing phase with well-trained models, adversarial poisoning attacks occur in the training phase, where the training data samples are modified by adversaries to mislead the training process to an erroneous model. Among different data poisoning attacks, label poisoning attack (e.g, [16, 21, 2, 3]) is crucial in semi-supervised learning, where the changes of labels may mislead the labeling of unlabeled data. For adversarial label poisoning attack, the adversary is only allowed to change the labels of a small number of training nodes. It can significantly reduce the performance of node classification than random label noise, because of the potential use of gradient information.

In the literature of graph semi-supervised learning, less attention has been paid to label poisoning attacks. Of particular relevance are the label-flipping attacks for the label propagation [14] and for the vanilla GCN model [35].
**Poisoning Attack on Label Propagation.** A unified framework has been proposed in [14] for graph semi-supervised learning with two algorithms to solve poisoning regression and classification tasks. The optimization objective is to maximize the error rate for classification with a combinatorial nature.
**Label-flipping Attack on GCNs.** LafAK [35] aims to flip the unnoticeable parts of the training labels to minimize the performance of GCN. Therefore, following the purpose of LafAK, the optimization problem can be written as

$$\begin{aligned} \min_{\delta} \quad & -L_{0-1}(\theta^*; A, X, y_u), \\ s.t. \quad & \theta^* = \arg\min_{\theta} L(\theta; A, X, \delta \odot y_l), \quad \|\delta - 1_{n_l}\|_0 \leq \epsilon n_l, \end{aligned} \tag{6}$$

where $y_l \in \{-1, +1\}^{n_l}$ and $y_u \in \{-1, +1\}^{n_u}$ are the labels of known and unknown nodes, respectively, $n_l$ and $n_u$ are the numbers of labeled and unlabeled nodes, $\delta \in \{+1, -1\}^{n_l}$ is the label perturbation, $1_{n_l}$ is an all-one vector, $\odot$ means Hadamard product, and $\epsilon$ is the ratio of label flipping.

## 3 Label Poisoning Attack Model

In this paper, we consider label poisoning attack on GCN-like models. Different from LafAK in [35] with label-flipping determined by gradient attack on the entire GCN model, we propose a novel label poisoning framework that delegates gradient attack solely to label propagation. This is inspired by recent advances on the equivalence of decoupled GCN and label propagation [4], for which label propagation followed by feature transformation performs as well as GCN models with coupled neighborhood aggregation and feature transformation.

Figure 1 presents our proposed framework on label poisoning attack. It consists of three components: (1) label propagation from labeled to unlabeled nodes to generate predictive pseudo-labels $\hat{y}_u$; (2) maximum gradient attack with changed labels to poison training data; and (3) GCN training with label poisoned data for node classification. In what follows, we explain the role of each component in detail.

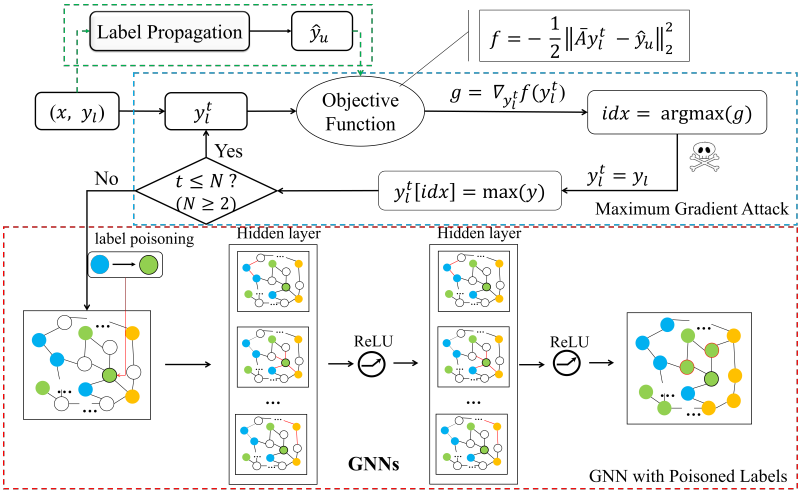


**Fig. 1.** The framework of our proposed label poisoning attack, which consists of label propagation to generate predictive labels, maximum gradient attack to poison data labels, and GNN training with poisoned labels. Specifically, in maximum gradient attack, the maximum gradients of the objective function are identified and the corresponding labels are poisoned, and it repeats several times ($N = 2$ is a good choice in practice). The label poisoned data are used for GNN training, misleading the trained model to reduce accuracy on unlabeled node classification.

### 3.1 Label Propagation

The goal of label propagation is to generate label prediction for unlabeled data, so as to be used in the maximum gradient attack.

We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of $n$ nodes, $\mathcal{E}$ is the edge set, and the adjacency matrix is represented by $A \in \{0,1\}^{n\times n}$. All nodes in the graph are associated with some node features, denoted by $X = [x_1; x_2; \ldots; x_n] \in \mathbb{R}^{n\times f}$ with $x_i \in \mathbb{R}^f$, and $y = [y_l; y_u]$ is the label set, including the known labels $y_l \in \mathbb{R}^{n_l}$ and the unknown labels $y_u \in \mathbb{R}^{n_u}$. The goal of node classification is to figure out $y_u$ with the knowledge of $X$, $y_l$, and $A$.

Let $S$ be a similarity matrix with $S_{ij} = \exp(-\gamma\|x_i - x_j\|^2)$ resulted from a Gaussian kernel, and $D$ be the degree matrix. Then, the graph Laplacian can be defined as $L = D - S$. Given $n_l$ labeled data and $n_u$ unlabeled data (usually $n_l < n_u$) with $n = n_l + n_u$ nodes, $S$ and $D$ are both $(n_l+n_u)\times(n_l+n_u)$ matrices. We split $S$ and $D$ into 4 sub-matrices $S = \begin{bmatrix} S_{ll} & S_{lu} \\ S_{ul} & S_{uu} \end{bmatrix}$, $D = \begin{bmatrix} D_{ll} & D_{lu} \\ D_{ul} & D_{uu} \end{bmatrix}$. The predicted labels via label propagation [38, 14] can be written as

$$\hat{y}_u = (D_{uu} - S_{uu})^{-1} S_{ul} y_l. \tag{7}$$

The label propagation takes the input features $X$ and some known labels $y_l$ to generate predictive labels $\hat{y}_u$, which are used to replace $y_u$ in the optimization problem of maximum gradient attack.

### 3.2 Maximum Gradient Attack

The aim of label poisoning attacks is to maximize the classification error rate by modifying a small fraction of known labels of data points at the training stage, that is, the optimization problem can be given by

$$\min_{y_l'} -\frac{1}{2}\left\|\bar{A}y_l' - (y_u \text{ or } \hat{y}_u)\right\|_2^2 + \lambda\|y_l' - y_l\|_0, \tag{8}$$

where $\bar{A}$ is the label propagation method which will be specified later, $y_l'$ represents the poisoned labels, the labels of unknown data is $y_u$ (ground truth) or $\hat{y}_u$ (predicted labels), $\|\cdot\|_0$ is the $\ell_0$-norm of a vector to ensure the number of poisoned labels is as small as possible, and $\lambda$ is to balance between the loss of label propagation and the number of poisoned labels.

In this section, we study the multi-class classification on graph convolution networks. The difference with the binary class labels is that the perturbed label $y_l'$ is changed to the class of another label directly instead of multiplying by a perturbation, because the input is multi-class data, such as $y \in \{0,1,2,3,4,5\}^n$. The label propagation method $\bar{A}$ can reuse those in GNN models specified in Section 2.1. In what follows, we consider three approaches:

*a.* **SM**: $\bar{A} = (D_{uu} - S_{uu})^{-1} S_{ul}$ as in [38, 14], and $\bar{A}y_l'$ means the class of the perturbed labels will be propagated from $y_l'$ to unobserved labels $y_u$.

b. **SK**: $\bar{A} = \hat{A}^K$ as in [30], where $\hat{A}$ is the normalized adjacency matrix and $K$ is the number of layers.
c. **SP**: $\bar{A} = \alpha(I - (1-\alpha)\hat{A})^{-1}$ as in [11, 37], which has been used to classify each node using information from the larger, adjustable neighborhood.

As $y_l'$ is an integer vector, Equation (8) is an integer program and in general challenge to derive a closed-form solution. To find a feasible solution, we employ a gradient descent method, namely maximum gradient (MG) attack, to generate an approximate solution in an iterative manner. MG attack takes known labels $y_l$ and the predictive labels $\hat{y}_u$ as the inputs, and outputs the indices of labels in $y_l$ for label poisoning and the poisoned labels.

The following Algorithm 1 summarizes the specific process of MG attack.

**Algorithm 1:** Maximum Gradient Attack

**Input:** Graph structured data with $A$, $X$, $y_l$, and the budget of poisoned labels $c$
**Output:** The poisoned labels $y_l'$ with $\|y_l' - y_l\|_0 \leq c$

$y_l^0 = y_l$
**for** $t \leq N$ **do**
  Compute $\bar{A}$ with different label propagation strategies;
  Compute gradient: $g = \nabla_{y_l^t} f(y_l^t)$ where $f(y_l^t) := -\frac{1}{2}\left\|\bar{A}y_l^t - (y_u \text{ or } \hat{y}_u)\right\|_2^2$;
  Select top $c$ gradients in $g$ and identify the indices of them as a set $\mathcal{I}$;
  Set $y_l^t = y_l$;
  Modify $y_l^t[i]$ for all $i \in \mathcal{I}$ to the max label class;
  Update $y_l^{t+1} \leftarrow y_l^t$, $t \leftarrow t+1$;
**end**
**Return** $y_l' = y_l^N$

### 3.3 GCN Training with Poisoned Labels

The updated $y_l'$ is the output result, which is used to train the GCN model together with $A$ and $X$. The resulting prediction of $\hat{\hat{y}}_u$ will be compared with the ground truth $y_u$ to test the performance of poisoning attack.

## 4 Experiments

**Datasets:** For label propagation, we conduct experiments on a binary classification dataset called rcv1,[1] which is a benchmark dataset on text categorization. It is a collection of newswire articles produced by Reuters in 1996-1997. For graph

[1] Publicly available at https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/.

neural networks, we use cora_ml [15], citeseer [23], pubmed [17] and ms_academic [24] datasets on PT, DGCN and GCN models. Cora_ml, citeseer and pubmed are citation networks, where each node indicates a paper and edges represent corresponding citation relationships. Ms_academic is a co-author network in which nodes and edges represent authors and co-authors, respectively. The dataset statistics are summarized in the Table 1.

**Table 1.** Dataset statistics

| Dataset | Nodes | Features | Classes |
|---|---|---|---|
| rcv1 | 20,242 | 47,236 | 2 |
| cora_ml | 2,810 | 2,879 | 7 |
| citeseer | 2,110 | 3,703 | 6 |
| pubmed | 19,717 | 500 | 3 |
| ms_academic | 18,333 | 6,805 | 15 |

**Baseline:**
*Label Propagation.* In the poisoning attack on LP, the greedy and probabilistic methods are proposed by [14] for label-flipping in binary classification.

- *Greedy:* The greedy solver is to flip data labels one by one in a greedy manner. In this process, the label that decreases the objective function the most after flipping is to be flipped. Repeat this step several times until the number of flipped labels reaches the budget.
- *Probabilistic:* The probabilistic method considers the flipping actions that may be better in the long run but may be suboptimal at present. It assigns high probability to each best action, but still retains non-zero probability for other actions, so as to determine the most suitable label for flipping.

*Graph Convolution Network.* In the poisoning attacks on GCN models, there are two baseline methods with random noise added to the labels [4] and with gradient attack using LafAK [35].

- *Random:* For the propagation then training model, the add_noise method is to randomly transfer some samples of each class and change their labels to other classes of label.
- *LafAK:* The objective of LafAK is a non-differentiable bi-level optimization problem. An approximate closed-form of GCN model parameters is considered, and continuous surrogates of non-differentiable components are adopted to simplify the gradient-based optimization process.

**Experimental Configuration:** We first verify the effectiveness of the proposed maximum gradient attack method against the label propagation (LP) model in graph semi-supervised learning [38], and then against the state-of-the-art GNN models, such as GCN [10], graph attention network (GAT) [26], GraphSAGE

[7], SGCN [30], APPNP [11], and PT [4]. For each model, we use the default settings for the hyper-parameters. In the experiments of attack on GCNs, we use the Adam [9] optimizer with a learning rate of 0.01 to train all models for a maximum of 2000 epochs, and also, we terminate training if the validation loss does not decrease for 100 consecutive epochs, using an early stopping window size of 100.

## 5 Attack Performance Evaluation

**Hypotheses.** Our experiments aim to validate two hypotheses:

1). Attacking GNNs can be alternatively done on the equivalent LP via gradient-based methods, which means neural networks can be ignored when determining attacked labels.
2). In the presence of node features, graph structural information is not necessary for an effective poisoning attack.

In order to test these two hypotheses, we conduct the following experiments. First, we verify the effectiveness of the gradient-based attacks on both LP and GCN models. Second, we inspect the choices of parameters in optimizing Equation (8) by applying the proposed MG attacks to GCN models with three label propagation methods and with/without ground truth. Third, we compare our proposed MG attack with with LafAK, confirming the above two hypotheses that label poisoning of GCN can be achieved in the context of ignoring neural networks and only knowing node characteristics. Finally, we apply MG attacks to other state-of-the-art GNN models, such as GAT, GraphSAGE, SGCN, APPNP, and PT, and demonstrate that these two hypotheses are also valid for these GCN-like models.

**Effectiveness of Gradient-based Attacks.** Figure 2 presents the reduced accuracy of label prediction with different attack methods on label propagation (Figure 2(a)) and GCN models (Figures 2(b)-(d)) over different datasets. In each subfigure, x-axis is the number (or rate in percentage) of attacked labels, and y-axis represents the prediction accuracy of label propagation or trained GCN models with poisoned labels. Note that the lower the accuracy of prediction, the more effective the attack method.

In order to confirm the effectiveness of gradient-based attacks on label propagation, Figure 2(a) presents the comparison of greedy and probabilistic methods with the existing FGSM and PGD attacks and our proposed MG attack method over the dataset rcv1 for binary classification. It can be seen that with the increase of the number of flipped labels, the performance of the gradient-based attacks outperform the greedy and probabilistic methods.

Figures 2(b)-(d) compare the performance of adding random label noise (add_noise) and our MG attack on PT and APPNP models, as well as the sole MG attacks on the neural networks, specifically when the adversary attacks the pseudo labels after the label propagation so that the poisoned pseudo labels are input into the neural networks, i.e., the blue line PT_NN(MG), over datasets

cora_ml, citeseer, and ms_academic. According to the framework in Figure 1, we first generate attacked labels based on label propagation via Algorithm 1, and then train the graph neural networks on the poisoned pseudo-label data. Given the equivalence between PT and DGCN by the loss function [4], we also apply our proposed MG attack to DGCN, i.e., APPNP. As shown in Figures 2(b)-(d), our proposed MG attack is indeed effective for the DGCN model. That is, it confirms our hypothesis that attacking the equivalent LP model is effective as attacking the GNN models, and that the gradient-based attacks are more powerful than randomly adding label noise.

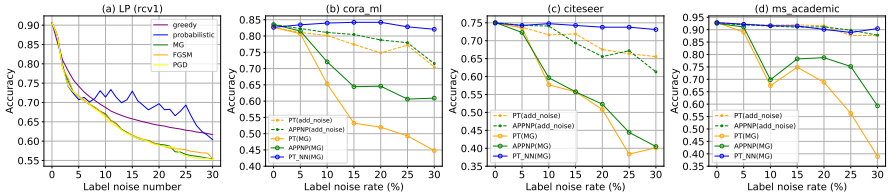


**Fig. 2.** (a) Accuracy performance comparison of the greedy, probabilistic, MG, FGSM and PGD attack methods on label propagation model over rcv1 dataset. (b)-(d) Comparison of the random add_noise and our maximum gradient attack methods on PT and APPNP models and only attack neural network on PT (PT_NN(MG)) on cora_ml, citeseer and ms_academic datasets.

**Label Propagation for Predicting $\hat{y}_u$.** The optimization problem in Equation (8) requires the knowledge of unknown labels either as the ground truth $y_u$ or as label predictions $\hat{y}_u$. For predicting $\hat{y}_u$, in our proposed framework in Figure 1, we adopt the label propagation method SK to generate label predictions, that is, $\hat{y}_u = \hat{A}^K y_l$, when the ground truth is unknown and the GCN model has not yet been trained. Figure 3 indicates the feasibility of this alternative $\hat{y}_u$ with different choices of label propagation strategies of $\bar{A}$ (SM, SK, SP) on attacking the GCN models. For example, SK_$y_u$ means $\bar{A} = \hat{A}^K$ and the ground truth $y_u$ are used in Equation (8). From Figure 3, we observe that using the predictive labels $\hat{y}_u$ results in more effective and stable attack performance.

**Label Propagation for MG Attacks with $\hat{y}_u$.** Our proposed MG attack method is mainly to solve the optimization problem in Equation (8), which involves the choices of label propagation methods in both MG attacks on the equivalent LP model with $\bar{A}$ and label predictions $\hat{y}_u$ resulted from LP algorithms. For label propagation, we consider three different choices of propagation strategies: SM, SK, SP, which indicate three label propagation methods $\bar{A}$ as mentioned in Section 3.2, respectively.

Figure 4 shows the attack performance with various combinations of three propagation methods for MG attacks and predicting $\hat{y}_u$. For instance, 'SK_SP' means $\bar{A} = \hat{A}^K$ is used for the MG attack and $\hat{y}_u = \alpha(I - (1-\alpha)\hat{A})^{-1} y_l$ for label prediction. It can be seen from Figure 4 that comparable performance can

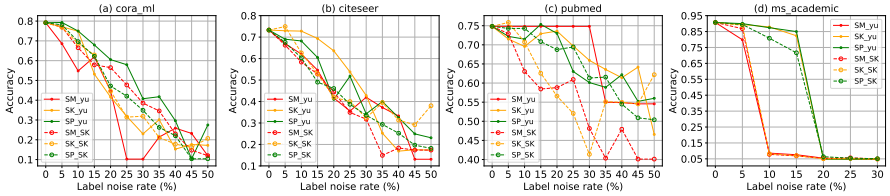


**Fig. 3.** The feasibility of substituting $\hat{y}_u = \hat{A}^K y_l$ for $y_u$. In the legend, the first part represents the choice of label propagation method $\bar{A}$, and the second part indicates if the ground truth $y_u$ or the predicted labels $\hat{y}_u = \hat{A}^K y_l$ are used.

be obtained by various propagation methods. Notably, SM_SM only uses data features for the MG attack without knowing ground truth, which validates the second hypothesis that LP-based gradient attack is also effective to poison GCN labels without the knowledge of graph structures and label ground truth.

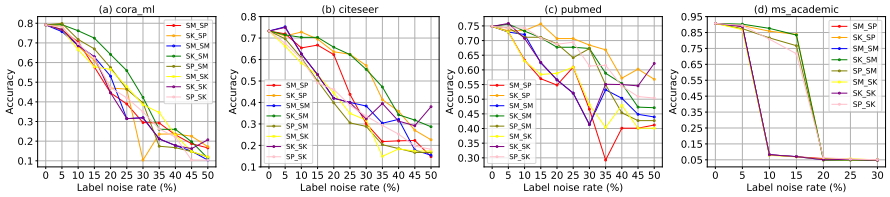


**Fig. 4.** Comparison of attack performance on GCN models with different combinations of label propagation methods $\bar{A}$ for the MG attack and for generating label predictions $\hat{y}_u$. In the legend, the first part represents the LP method of $\bar{A}$, and the second one is the propagation method to generate predicted labels $\hat{y}_u$ with SM: $(D_{uu} - S_{uu})^{-1} S_{ul}$; SK: $\hat{A}^K$; and SP: $\alpha(I - (1-\alpha)\hat{A})^{-1}$.

**Comparison of MG Attack and LafAK.** We compare the attack performance of our proposed MG attack with the state-of-the-art LafAK method on the GCN models to validate the above two hypotheses. While LafAK can be seen a white-box attack where neural network parameters are accessible to the adversary, our proposed method is a black-box attack such that the adversary does not require the knowledge of neural network parameters to make an effective attack. In our proposed MG attack, we apply three different label propagation methods of $\bar{A}$, marked by SM, SK, APPNP, and the predicted labels $\hat{y}_u = \hat{A}^K y_l$. LafAK is a label-flipping attack through maximizing the loss of the entire GCN model with neural network. It selects two classes of labels with the largest number in the dataset to flip. There is a limitation: When the labels of these two nodes are all flipped to the opposite class, the attack no longer takes effect, as shown in Figure 5 where the 'LafAK_o' curves experience error floors. To rectify this and ensure the fairness of comparison, we extend LafAK to all training datasets

(called 'LafAK_c'), and modify the attacked labels to the maximum label class in the dataset as our MG attack does. From the comparison in Figure 5, we conclude that:

1). Our proposed MG attack (on the equivalent LP only without considering neural network model parameters) is as effective as LafAK (on the entire GCN models with neural model parameter involved). Therefore, this confirms the first hypothesis that attacking GCN models can be alternatively done by attacking the equivalent LP model.
2). For our proposed MG attacks with different label propagation methods $\bar{A}$, the attack performances are comparable, so that $\bar{A} = (D_{uu} - S_{uu})^{-1}S_{ul}$ without topological structure still yields an effective attack. This confirms the second hypothesis that graph structural information is not necessary.

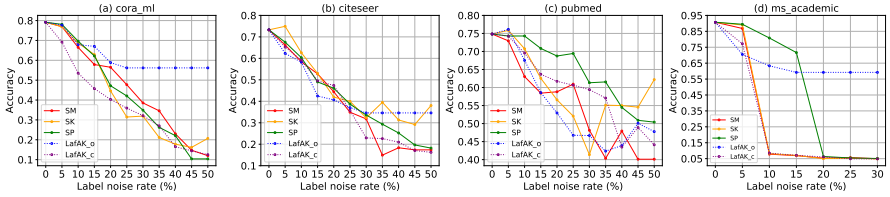


**Fig. 5.** Comparison between MG attacks and LafAK against GCN models. In the legend of this figure, SM, SK, SP means three different label propagation methods of $\bar{A}$, LafAK_o is the original LafAK, and LafAK_c is an improved version of LafAK_o with the limitation of target flipping labels fixed.

**Transferability of MG Attacks.** In practical scenarios, the adversary may not know which GNN model is under training. So, to test the possibility of applying attacks learned for one model to other models, we consider the transferability of our MG attack. Recall that our MG attacks depend only on the input futures and the known labels, and are irrelevant to the GNN models. As such, we directly apply the poisoned labels from our MG attack to state-of-the-art GNNs. Figure 6 presents the transferability to the GCN-like models, such as GAT and GraphSAGE. The results indicate that the our proposed MG attacks can be successfully transferred to these two models.

**Robustness of Different GNN Models against MG Attacks.** Figure 7 gives a more comprehensive study of the robustness of state-of-the-art GNN models against our proposed MG attack over different datasets. In particular, the MG attack with 'SM_SK' is employed, that is $\bar{A} = (D_{uu} - S_{uu})^{-1}S_{ul}$ for MG attacks and $\hat{y}_u = \hat{A}^K y_l$ for pseudo-label prediction. According the experiments, we have the following observations.

1). The robustness of APPNP always outperforms other models under MG attacks. It is probably because APPNP can aggregate information from large, adjustable neighborhood for classifying each node.

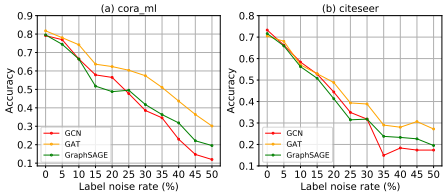


**Fig. 6.** Transferability of our MG attack to GAT and GraphSAGE models.

2). Although SGCN has comparable accuracy with clean data as other models, the linearity makes it more vulnerable to label poisoning attacks, because of the removal the nonlinear activation functions between GCN layers.
3). PT has a similar accuracy performance as GCN and GraphSAGE with both clean and poisoned data, and their robustness against the MG attacks varies across different datasets.

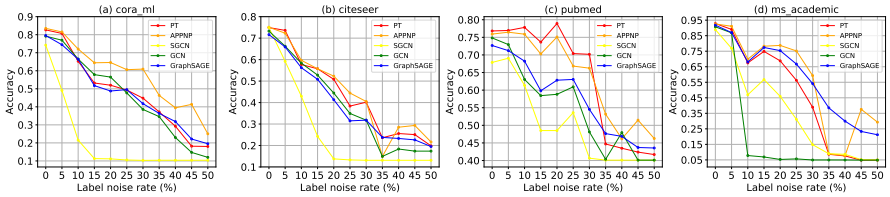


**Fig. 7.** The robustness of state-of-the-art GNN models (GCN, GraphSAGE, SGCN, APPNP, PT) against our MG label poisoning attack with 'SM_SK' label propagation methods over different datasets.

## 6 Conclusion

In this paper, we studied adversarial label poisoning attacks on GNNs and proposed a framework to attack GCN-like models through a maximum gradient attack method on the equivalent LP models. The proposed method can effectively attack the GCN-like models while avoiding the neural network computation, thereby reducing the node classification performance of the GCNs. Extensive experiments showed that, our proposed label poisoning attacks only on LP is as effective as the state-of-the-art attacks (e.g., LafAK), and is transferable to the GCN-like models. Notably, our method does not require the knowledge of graph structural information in the presence of node features.

**Acknowledgment.** GL thanks the Department of Computer Science, University of Liverpool for the support of the attendance to the conference. XH is supported by the EPSRC under project [EP/T026995/1].

## References


1. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering **31**(5), 833–852 (2018)
2. Dai, E., Aggarwal, C., Wang, S.: NRGNN: Learning a label noise-resistant graph neural network on sparsely and noisily labeled graphs. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 227–236 (2021)
3. Dai, E., Jin, W., Liu, H., Wang, S.: Towards robust graph neural networks for noisy graphs with sparse labels. arXiv preprint arXiv:2201.00232 (2022)
4. Dong, H., Chen, J., Feng, F., He, X., Bi, S., Ding, Z., Cui, P.: On the equivalence of decoupled graph convolution network and label propagation. In: Proceedings of the Web Conference 2021. pp. 3651–3662 (2021)
5. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: The World Wide Web Conference. pp. 417–426 (2019)
6. Gao, J., Zhang, T., Xu, C.: Graph convolutional tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4649–4659 (2019)
7. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035 (2017)
8. Jin, W., Li, Y., Xu, H., Wang, Y., Tang, J.: Adversarial attacks and defenses on graphs: A review and empirical study. arXiv e-prints pp. arXiv–2003 (2020)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)
11. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: International Conference on Learning Rep resentations, ICLR, 2019 (2019)
12. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph lstm. In: European Conference on Computer Vision. pp. 125–143. Springer (2016)
13. Liu, C.Y., Zhou, C., Wu, J., Hu, Y., Guo, L.: Social recommendation with an essential preference space. In: Thirty-second AAAI conference on artificial intelligence (2018)
14. Liu, X., Si, S., Zhu, X., Li, Y., Hsieh, C.J.: A unified framework for data poisoning attack to graph-based semi-supervised learning. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. pp. 9780–9790 (2019)
15. McCallum, A.K., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval **3**(2), 127–163 (2000)
16. Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 27–38 (2017)
17. Namata, G., London, B., Getoor, L., Huang, B., EDU, U.: Query-driven active surveying for collective classification. In: 10th International Workshop on Mining and Learning with Graphs. vol. 8, p. 1 (2012)

18. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
19. Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Guerrero, R., Glocker, B., Rueckert, D.: Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. Medical image analysis **48**, 117–130 (2018)
20. Rieck, B., Bock, C., Borgwardt, K.: A persistent weisfeiler-lehman procedure for graph classification. In: International Conference on Machine Learning. pp. 5448–5458. PMLR (2019)
21. Rosenfeld, E., Winston, E., Ravikumar, P., Kolter, Z.: Certified robustness to label-flipping attacks via randomized smoothing. In: International Conference on Machine Learning. pp. 8230–8241. PMLR (2020)
22. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008)
23. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
24. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. Relational Representation Learning Workshop , R2L (2018)
25. Sun, L., Dou, Y., Yang, C., Wang, J., Yu, P.S., He, L., Li, B.: Adversarial attack and defense on graph data: A survey. arXiv preprint arXiv:1812.10528 (2018)
26. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. (2018)
27. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
28. Wang, X., Gupta, A.: Videos as space-time region graphs. In: Proceedings of the European conference on computer vision (ECCV). pp. 399–417 (2018)
29. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions On Graphics (TOG) **38**(5), 1–12 (2019)
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, J., Jiang, H., Ding, X., Konda, A., Han, J., Zhang, Y., Li, Q.: Learning differential diagnosis of skin conditions with co-occurrence supervision using graph convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 335–344. Springer (2020)
32. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
33. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 974–983 (2018)
34. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. Advances in neural information processing systems **31** (2018)

35. Zhang, M., Hu, L., Shi, C., Wang, X.: Adversarial label-flipping attack and defense for graph neural networks. In: 2020 IEEE International Conference on Data Mining (ICDM). pp. 791–800. IEEE (2020)
36. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. Advances in neural information processing systems **31** (2018)
37. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)
38. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)