Poster: Unlearnable Graph: Protecting Graphs From Unauthorized Exploitation

Yixin Liu*, Chenrui Fan†, Pan Zhou† and Lichao Sun*
* Lehigh University, Bethlehem, PA, USA
† Huazhong University of Science and Technology, Wuhan, Hubei, China
{yila22, lis221}@lehigh.edu, {chenrui_fan, panzhou}@hust.edu.cn

***Abstract*—While the use of graph-structured data in various fields is becoming increasingly popular, it also raises concerns about the potential unauthorized exploitation of personal data for training commercial graph neural network (GNN) models, which can compromise privacy. To address this issue, we propose a novel method for generating unlearnable graph examples. By injecting delusive but imperceptible noise into graphs using our Error-Minimizing Structural Poisoning (EMinS) module, we are able to make the graphs unexploitable. Notably, by modifying only 5% at most of the potential edges in the graph data, our method successfully decreases the accuracy from 77.33% to 42.47% on the COLLAB dataset.**

I. Introduction

The abundance of data has led to the successful implementation of deep learning, which allows the integration of artificial intelligence (AI) into various domains. However, With the increasing availability of publicly accessible data, concerns have risen about the unauthorized exploitation of data. Many commercial AI models are trained using personal data that is unknowingly collected from the internet, raising questions about the potential misuse of this data for commercial or even illegal gain and also posing a significant threat to individuals' privacy, security, and copyright.

The threat of unauthorized data exploitation has made it imperative to develop defensive approaches. Recent studies have been focusing on developing *Unlearnable Example* [1, 2, 3]. These methods aim to make the original data *unlearnable* by adding imperceivable but delusive perturbations to data samples, resulting in deep learning models trained on the perturbed dataset having extremely low prediction accuracy.

Previous studies on unlearnable examples have primarily focused on the vision domain. However, as the use of graph data structures becomes more prevalent, particularly in regard to privacy and security, it is important to explore the potential vulnerability of unauthorized graph exploitation. As far as we know, unlearnable graphs, i.e., unlearnable examples on graph data, have not been explored yet. In this paper, we aim to answer the question of how to make structured graph data unlearnable by a wide range of GNN models.

To tackle these issues, we propose the *Adaptive GradArgMin* method to craft error-minimizing structural perturbation based on the gradient information. The *Adaptive GradArgMin* selects a set of edges that cause the maximum gradient change and conducts the flipping operation in the adjacent matrix. To achieve a good balance between invisibility and effectiveness under limited manipulation budgets, we design an adaptive constraint strategy by considering both vertex-based and edge-based information. The perturbed graph maintains invisible compared to the original graph under visual inspections, which ensures the utility of the data for other purposes while making the modified data unexploitable by ML models.

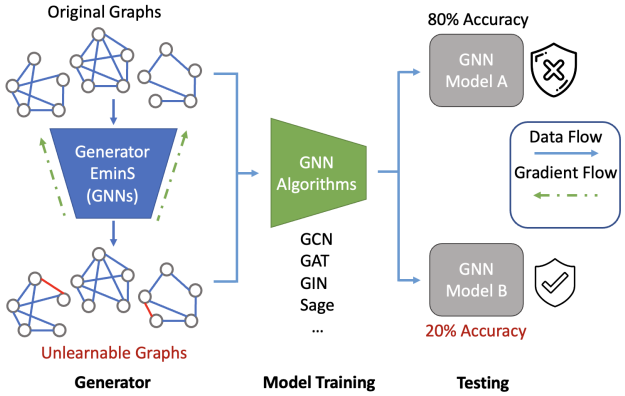

Fig. 1: An illustration of motivation of Unlearnable Graph. Existing vision-based solutions fail to inject delusive patterns into more challenging data structure-graph due to their discrete property. In this paper, we propose an Error-Minimizing Structural Poisoning to achieve efficient and effective data protection for graphs.

II. Assumptions and Problem Formulation

Assumptions on Defender's Capability. We assume that the data owner has full access to a portion of graph data used to train a model by unauthorized data exploiters. However, the defender could not interfere with the model selection and the training procedure of the unauthorized users.

Objective. Given a clean graph training dataset $\mathcal{G}_c = \{G_i, y_i\}_{i=1}^N$, our goal is to craft an unlearnable version of the training dataset $\mathcal{G}_u = \{\hat{G}_i, y_i\}_{i=1}^N$ such that the models trained on the $\mathcal{G}_u$ have poor performance on the clean testing set $\mathcal{G}_t$. The task can be formulated into a bi-level optimization as follows:

$$\begin{aligned} &\max_{\delta_i \preceq \mathrm{c}} \mathop{\mathbb{E}}_{(\mathrm{G}_i, y_i)\sim\mathcal{G}_\mathrm{t}} \left[\mathcal{L}\left(f_{\theta^*}(\mathrm{G}_i), y\right)\right], \\ &\text{s.t.}\theta^* = \arg\min_{\theta} \sum_{(\mathrm{G}_i, y_i)\in\mathcal{G}_\mathrm{u}} \left[\mathcal{L}\left(f_\theta\left(\mathrm{G}_i \oplus \delta_i\right), y_i\right)\right]. \end{aligned} \tag{1}$$

where $\oplus$ denotes the application of perturbations of node features or topology structure on the original graph G_i, and $\preceq$ represents the budget constraints relationship.

III. Proposed Methodology

With the problem formulation above, we design the following technique to generate Error-Minimizing Structural noise, which is effective and imperceptible against unauthorized exploitation.

The Min-min Optimization. To tackle the intractable bi-level problem in Eq. 1, an approximated *min-min* optimization process is proposed [3] to first learn a noise generator and leverage it to conduct noise generation. The major motivation is to iteratively craft noise that can trick the models trained on the poisoned data. The problem is also a bi-level optimization problem, with two levels of minimization. The inner level is a constrained optimization problem that finds the noise that is bounded by certain constraints and minimizes the model's classification loss. The outer level is another minimization problem that finds the parameters that also minimize the model's classification loss.

Crafting Delusive Edges with Adaptive GradArgMin. The core of our method is to take gradients with respect to the adjacent matrix A to obtain the gradient for any edge in the potential edge space ($\mathcal{V} \times \mathcal{V}$) regardless of its existence. For one selected edge (u, v), we conduct the discrete version of the gradient descend by deleting existing edges with a positive gradient or adding non-exist edges with a negative gradient. Given the modification constraint of edges c and the element $\alpha_{u_t,v_t} \in A$, we obtain a set of edges via a greedy selection:

$$\{u_t, v_t\}_{t=1}^{c} = \underset{\{u_t,v_t\}_{t=1}^{c}}{\arg\max} \sum_{t=1}^{c} |\frac{\partial \mathcal{L}}{\partial \alpha_{u_t,v_t}}|. \tag{2}$$

After that, modifications are performed by sequentially modifying these edges in the way that is most likely to reduce the loss function. Note that we stop the modification process until we find all the gradients for existing edges are negative, or the ones for non-exist edges are positive, in which case no perturbation for decreasing the objective is possible.

Adaptive Constrains. To ensure that we modify *sufficient and essential* edges in each graph for creating *delusive and invisible* patterns, we devise a mixed type of constraint based on *vertex-based* and *edge-based* information. We consider two types of perturbation ratios, r_V and r_E, which refer to the entire edge space ($V \times V$) or existing edges (E), respectively. The number of edges to be modified is constrained by both coefficients. Our constraints are effective and flexible, as they allocate more budget to larger graphs, ensuring the overall imperceptibility and effectiveness of our method.

IV. Experiments

In order to evaluate the effectiveness of our proposed method, we conducted experiments on six benchmark graph classification datasets(MUTAG, ENZYMES, PROTEINS, IMDB-B, IMDB-M, and COLLAB) across four common GNN architectures(GCN, GAT, GIN, and GraphSage), and make comparisons between random and error-maximizing noise. The results on the PROTEINS and IMDB-M datasets are reported in Figure 2.

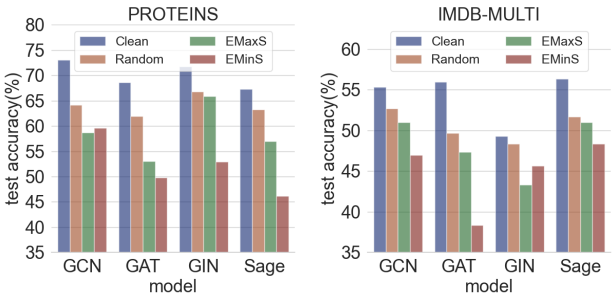

Fig. 2: A comparison among different methods on PROTEINS and IMDB-MULTI datasets.

Despite the good performance of EMinS noise in degrading the models' test accuracy, we visualize the graphs before and after perturbation in Table I to demonstrate the imperceptibility of our noise. From the visualizations, we can observe that for the majority of graphs, there are few visual discrepancies between the original and modified ones.

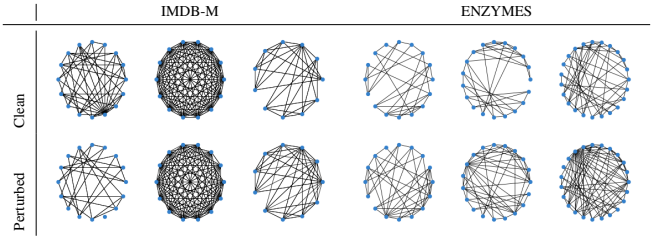

TABLE I: Graph visualizations on IMDB-MULTI and ENZYMES datasets (the first row are clean graphs, and the second row are the graphs we generated with EMinS).

V. Conclusion

In this paper, we are the first group that proposes a novel method for minimizing errors in structural poisoning for generating unlearnable graphs. Our method explores invisible noise to prevent GNN models from exploiting graph data freely. We verify our method by conducting experiments on six benchmark graph datasets, and the extensive experimental results show that our method can be applied effectively to various GNN architectures. This study represents an important first step in safeguarding personal graph data from being exploited by GNN models.

References

[1] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv preprint arXiv:2103.02683*, 2021.

[2] Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojtek Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *NeurIPS*, 2021.

[3] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *arXiv preprint arXiv:2101.04898*, 2021.

Unlearnable Graph: Protecting Graphs From Unauthorized Exploitation

Yixin Liu[1], Chenrui Fan[2], Pan Zhou[2], Lichao Sun[1]
[1]Leigh University, [2]Huazhong University of Science and Technology

INTRODUCTION

Research Motivation:

- The increasingly popular use of graph-structured data in various fields.
- Raising privacy and security concerns about the potential unauthorized exploitation of personal data.
- Recent studies of unlearnable examples in the vision domain.

Our Work:

- Injecting delusive but imperceptible **E**rror **Min**imizing **S**tructural noises **(EMinS)** into the graph to prevent the personal graph data from unauthorized training.
- **The first** step in safeguarding personal graph data from being exploited by GNN models.

OBJECTIVE

- To degrade the model's performance with a bounded modification on the training graph data.

DEFEND CAPACITY

- The defender has full access to a portion of the training data.
- The defender could not interfere with the unauthorized users' model selection and training procedure.

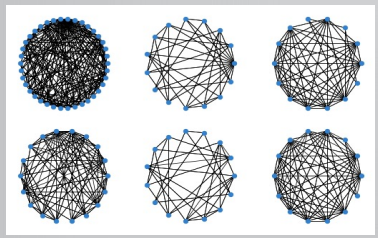

Table1. The visualization of clean (first row) and perturbed (second row) graphs from IMDB-M dataset.

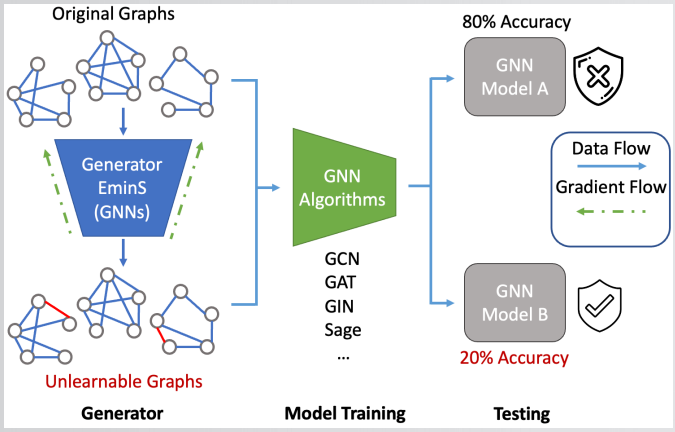

Fig 1. An illustration of the motivation for Unlearnable Graph.

METHODOLOGY

The Min-Min Optimization

- We suppress the gradient information on graphs by crafting low-error examples through a bi-level min-min optimization.

The GradArgMin Method

- We take gradients with respect to the adjacent matrix to obtain the gradient of any edges in the potential edge space, whether it exists or not.
- We rely on the gradient information and apply a greedy approach to flip the edges, which will most likely reduce the loss.

The Adaptive Budget

- We apply an adaptive budget that scales according to the size of the graph.

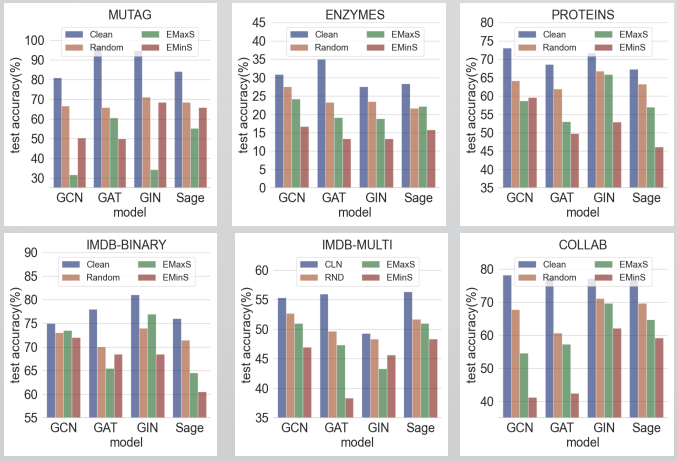

Fig 2. The experiment results across different models on various datasets.

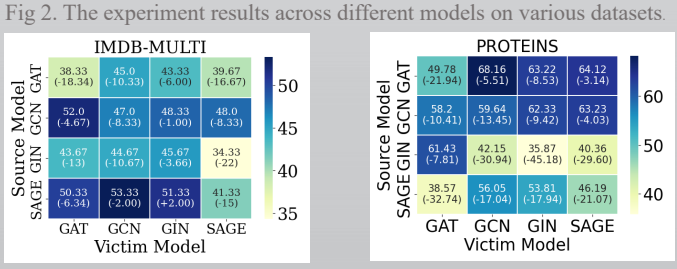

Fig3. The transferability study on the PROTEINS/IMDB-M dataset. The test accuracy and loss on clean data are reported.

EVALUATION SETUP

Datasets:

- 3 Bio-information Datasets: MUTAG, ENZYMES, PROTEINS
- 3 Social Network Datasets: IMDB-B, IMDB-M, COLLAB

Source Models:

- 4 Commonly Used GNNs: GCN, GAT, GIN, GraphSage

Compared Methods:

- Clean, Random, EMaxS (Error Maximizing noise), EMinS(Ours),

RESULTS

- Effective in degrading various models' performance
- Imperceptible on the majority of the datasets
- Good transferability across models on different datasets

FUTURE DIRECTIONS

- The current setting assumes that we have full knowledge of the architectures. But as our transferability study gives a satisfactory result, generating black-box unlearnable graphs may be flexible.
- The visualizations show that our method suffers from perceptibility problems on sparse graphs(e.g. MUTAG dataset). One future direction may be further improving the imperceptibility of sparse graphs.
- Further studies are needed to determine whether our method is sensitive to dataset size.