Rethinking Label Poisoning for GNNs: Pitfalls and Attacks

Vijay Lingam[1], Mohammad Sadegh Akhondzadeh[1,2], Aleksandar Bojchevski[2]
[1]CISPA Helmholtz Center for Information Security, [2]University of Cologne

ICML International Conference On Machine Learning

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

University of Cologne

TL;DR

- Label poisoning for GNNs is plagued by serious evaluation pitfalls.
- Existing attacks render ineffective post fixing these fallacies.
- We introduce two new simple yet effective family of attacks that are significantly stronger (up to **~8**%) than previous strongest attacks.

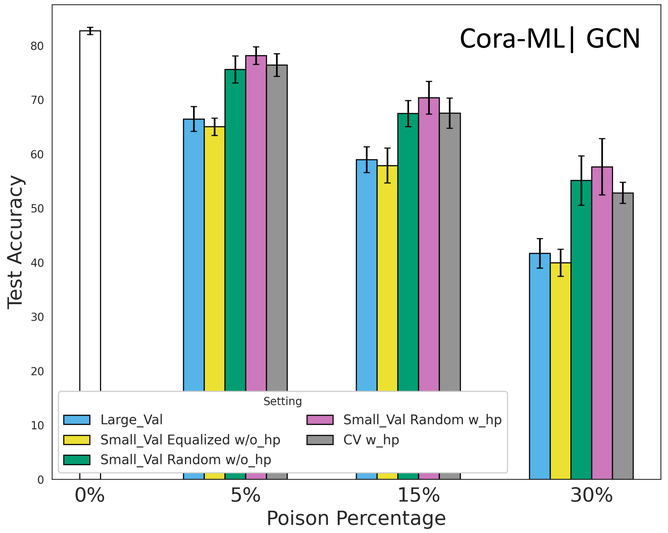
Motivation

GNNs have wide range of applications including critical ones.

Label poisoning poses a distinct threat as training data can be compromised.

Existing attacks are not effective; do better attacks exist?

Existing attacks are not as powerful as claimed



1. Large validation set
2. Class equalised splits
3. Hyper-param tuning
4. Clean Validation set
5. Missing stdev

Fixing the above pitfalls leads to a massive reduction in LafAK's performance (previous strongest attack).

Threat Model

Flip a small fraction of labels to decrease test acc. Results in a difficult bi-level optimization problem for which we propose different relaxations.

Baselines

Heuristic-based: Random (**RND**), Degree (**DEG**)

Learning-based: LP, LafAK (**LFK**), **MG**

Linear surrogate attacks

Linearize the classifier and compute the optimal weights in closed-form

$$\min_{\boldsymbol{H}\in\{0,1\}^{n\times C},\ \boldsymbol{b}\in\{0,1\}^{n}} \mathcal{L}(\boldsymbol{Y}_u, \widetilde{\boldsymbol{Y}}_u)$$

$$\widehat{\boldsymbol{Y}}_l = \boldsymbol{b}\odot\boldsymbol{H} + (\boldsymbol{1}_N - \boldsymbol{b})\odot\boldsymbol{Y}_l \quad \text{\# poisoned labels}$$

$$\boldsymbol{H} \neq \boldsymbol{Y}_l$$

$$\widetilde{\boldsymbol{Y}}_u = \widehat{\boldsymbol{X}}_u\widetilde{\boldsymbol{X}}_l\widehat{\boldsymbol{Y}}_l \quad \text{\# compute predictions}$$

$$\boldsymbol{H}\boldsymbol{1}_C = \boldsymbol{1}_N \quad \text{\# one-hot constraint}$$

$$\boldsymbol{b}^T\boldsymbol{1}_N \leq \epsilon N \quad \text{\# enforces budget}$$

$$\widetilde{X} = (\widehat{X}^T\widehat{X} + \lambda I)^{-1}\widehat{X}^T \quad \text{\# Closed form solution of LR}$$

Variant-1: SGC surrogate $\widehat{X} = \widehat{A^2}\,X$

Variant-2: NTK surrogate $\widehat{X} = NTK - Kernel$

Meta attacks

Meta gradients w.r.t. labels by backpropagating through the unrolled inner optimization.

Final poisoned labels are constructed as follows:

$$\widehat{\mathbf{Y}}_l = \mathbf{b}\odot\mathbf{H} + (1-\mathbf{b})\odot\mathbf{Y}_\mathbf{l}$$

where $\mathbf{H} = GumbelSoftmax(\widetilde{\mathbf{H}});\quad \widetilde{\mathbf{H}}\ in\ \mathbb{R}^{N\times C}$

$$\mathbf{b} = top_k(\widetilde{\mathbf{b}});\quad \widetilde{\mathbf{b}} \in \mathbb{R}^N$$

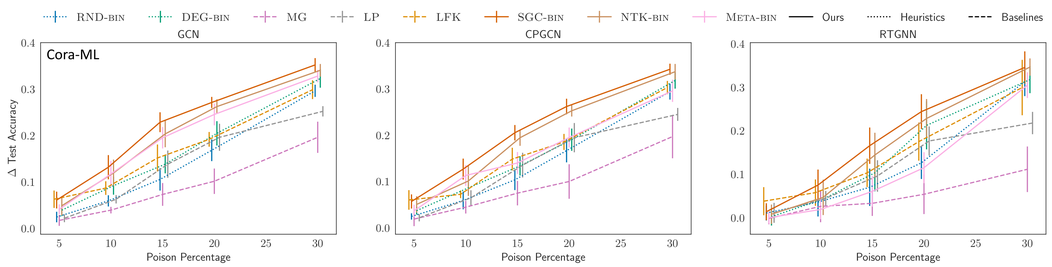
Note: since top_k is not differentiable, we apply soft-top-k followed by k-subset selection.

Gumbel-softmax trick to approximate 0-1 loss

To make the 0-1 loss differentiable, we propose the following simple alternative:

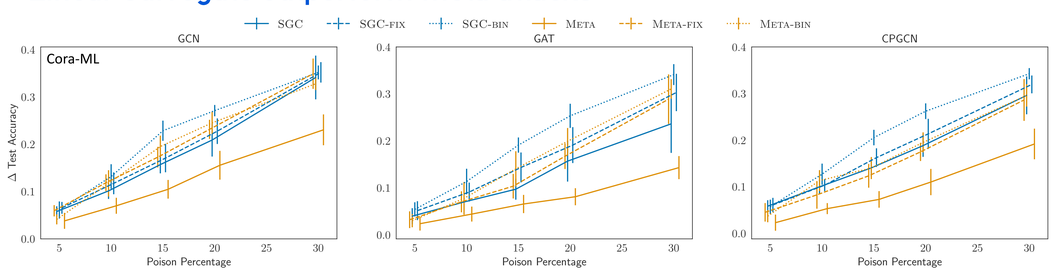
$$GS_{loss} = \frac{1}{|u|}\sum GumbelSoftmax(\widehat{Y}_u)\odot Y_u$$

Our proposed attacks significantly outperform baselines



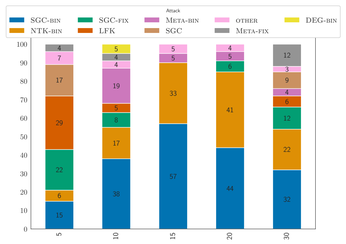
All our attacks (in solid), particularly SGC-BIN, outperform baseline attacks with max gains of up to ~13%.

Linear surrogate outperform meta attacks



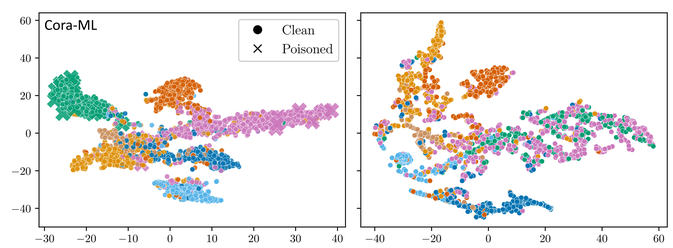
Different variants of the attacks that we propose. Binary variants are best on average.

Strongest attack per split



Different variants of our linear attacks win most often.

t-SNE of the logits before and after poisoning



Poisoning only a handful of training labels disrupts the learned representations compared to clean model representations.

Key takeaways

* Faithfully simulating the defender is crucial to evaluate the efficacy of an attack.
* Simple label poisoning attacks (especially the binary variants) are surprisingly powerful.
* Our findings highlight the need to further study label poisoning attacks as well as develop defences.