Malicious Node Injection: A Transferable Adversarial Attack on GNN Fairness

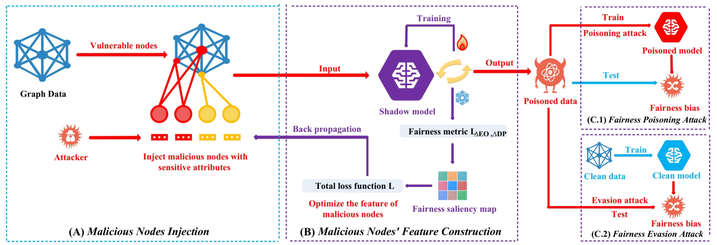
Haotian Zhang · He Huang* · Shuang Cui · Yu-E Sun
Soochow University, Suzhou, China | *Corresponding author

Motivation

Fairness-aware GNNs can still produce unequal outcomes under adversarial manipulation. Prior fairness attacks often edit existing edges; NIFA [1] injects nodes but is limited to poisoning.

Can node injection also compromise a clean, already-trained model at inference, and transfer across architectures? UFA addresses both settings using the same perturbed graph, including multi-class tasks with multiple sensitive groups.

Method



UFA selects vulnerable targets, optimizes injected-node features on a shadow GNN, and reuses the graph for poisoning or evasion.

Input: graph topology, features, observed labels and sensitive attributes. Output: a graph augmented with fewer than 0.2% malicious nodes. Original nodes, features and existing edges are preserved; only new nodes and their incident edges are added.

Boundary-sensitive node selection

A DropConnect GCN averages 20 stochastic predictions. The score combines a small top-two class margin with high prediction entropy, then applies min–max normalization:

$$Q = \left(1 - (p_{(1)} - p_{(2)}) - \sum_c p_c \log p_c\right)_{\text{nor}}$$

Select the top 40% of candidates within each sensitive group. Randomly assign injected nodes to groups and connect them to sampled targets in the matching group.

The gray-box attacker cannot access victim parameters or architecture. Optimization uses available training information, with no test feedback; injected nodes are excluded from evaluation.

Reference & code

[1] Luo et al. Are Your Models Still Fair? Fairness Attacks on Graph Neural Networks via Node Injections. NeurIPS, 2024.

Code (paper): github.com/pikalooo/ufa

Fairness saliency and feature optimization

The shadow GNN produces class-probability vectors. Its differentiable fairness loss measures dispersion of group means, globally (DP) and conditional on the true class (EO). A group saliency map averages input-feature gradients:

$$H_s = \frac{1}{|V_s|} \sum_{v \in V_s} \frac{\partial(\mathcal{L}_{\Delta\text{EO}} + \mathcal{L}_{\Delta\text{DP}})}{\partial X_v}$$

$$\mathcal{L}_f = -\frac{1}{|S|R} \sum_{s \in S} \|H_s - \bar{H}\|_2^2, \quad \bar{H} = \frac{1}{|S|} \sum_{s \in S} H_s$$

$$\mathcal{L} = \alpha \mathcal{L}_f - \mathcal{L}_{\text{CF}} + \mathcal{L}_U$$

Minimizing the objective separates sensitive-group saliency maps. The feature-dispersion term $\mathcal{L}_{CF}$ encourages divergent injected groups; training cross-entropy $\mathcal{L}_U$ penalizes utility loss. Here S is the group set, V_S its nodes, and R the saliency dimension.

Transferability and its condition

Lemma. 4. 1: For the same graph input, if the objective functions of two models exhibit similar local behavior within a neighborhood of the input and are differentiable with respect to the input features, then their input-gradient-based saliency maps are close.

Lemma 4.1 gives a sufficient local condition: differentiable models with similar local fairness responses have close input-gradient saliency maps. This is conditional, rather than a guarantee for every GNN. On Pokec-Z, cross-model cosine similarities to GCN range from 0.8409 to 0.9569.

Experiments

Five graphs: Pokec-Z/N, DBLP, AMiner-S/L; six standard and three fairness-aware GNNs. Labeled nodes use a 50/25/25 train/validation/test split; experiments repeat 20 times. Higher ΔEO/ΔDP means more bias; ACC measures utility.

Poisoning: binary and multi-class tasks

GCN / dataset	ACC (%)	ΔEO (%)	ΔDP (%)
Pokec-Z: clean	71.30	4.61	5.97
Pokec-Z: UFA	71.25	16.87	18.94
AMiner-L: clean	95.54	5.77	10.55
AMiner-L: UFA	95.36	9.83	12.73

AMiner-L has 3 classes and 7 sensitive groups. All values transcribed from Tables 1 and 3.

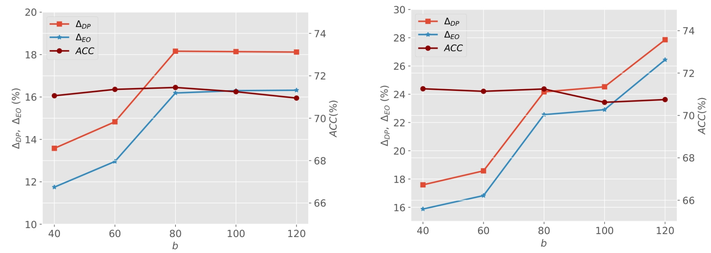
Evasion on Pokec-Z

GCN attack	ACC (%)	ΔEO (%)	ΔDP (%)
Clean	71.31	5.12	7.08
FA-GNN	71.09	10.33	12.03
UFA	70.62	**22.91**	**24.53**

Table 2 excerpt. NIFA does not support evasion. ΔEO/ΔDP values are percentages.

UFA raises ΔDP by 17.45 percentage points while ACC falls by 0.69 points. The same vulnerability affects fairness-aware models: FairGNN ΔDP rises from 1.72% to 30.60% on Pokec-Z. Utility preservation varies: DBLP/GCN ACC falls from 96.12% to 89.52%.

Budget sensitivity



Original Figure 3(a,d): poisoning (left), evasion (right), Pokec-Z / GCN. More injected nodes generally increase disparity; ACC also declines.

Conclusion

Small injections can undermine group fairness across GNN architectures and attack timings. Accuracy alone can miss these shifts; fairness needs explicit robustness auditing.

Scope: five node-classification benchmarks and a gray-box threat model. Transfer may weaken for models with different local responses or stronger defenses.