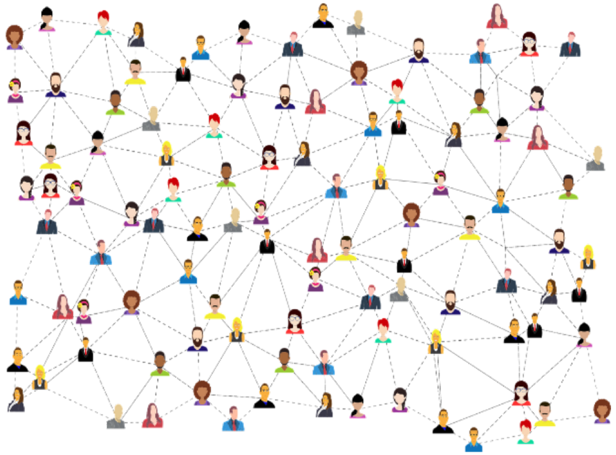
GraphMI: Extracting Private Graph Data from Graph Neural Networks

Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, Enhong Chen

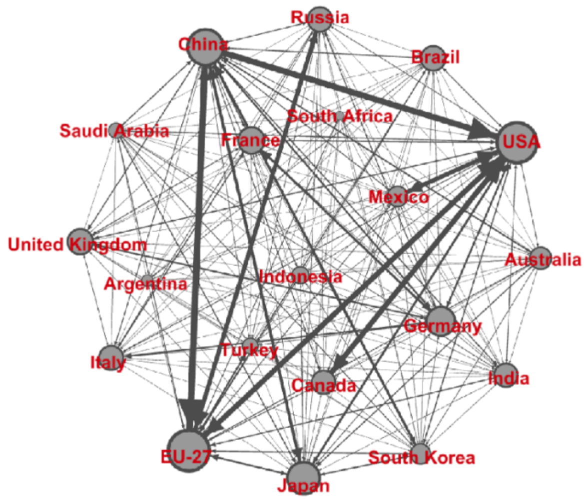
IJCAI 2021 MONTREAL

Graph Data and Graph Neural Networks

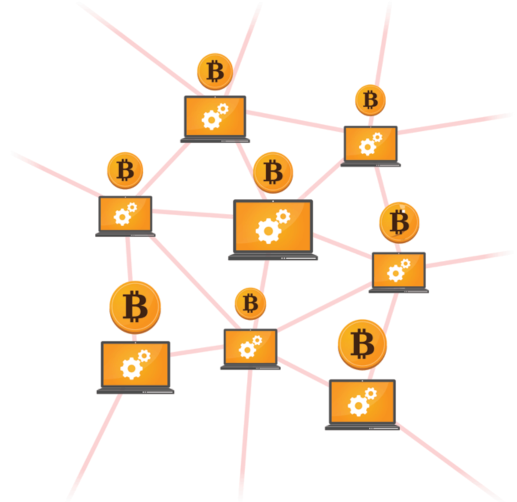
- Graphs are widely used to model complex interactions between entities
- Many graphs encode sensitive relational data



Social Network



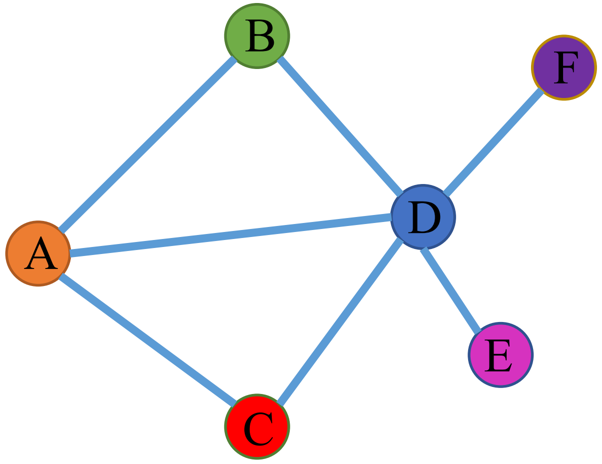
Trade Network

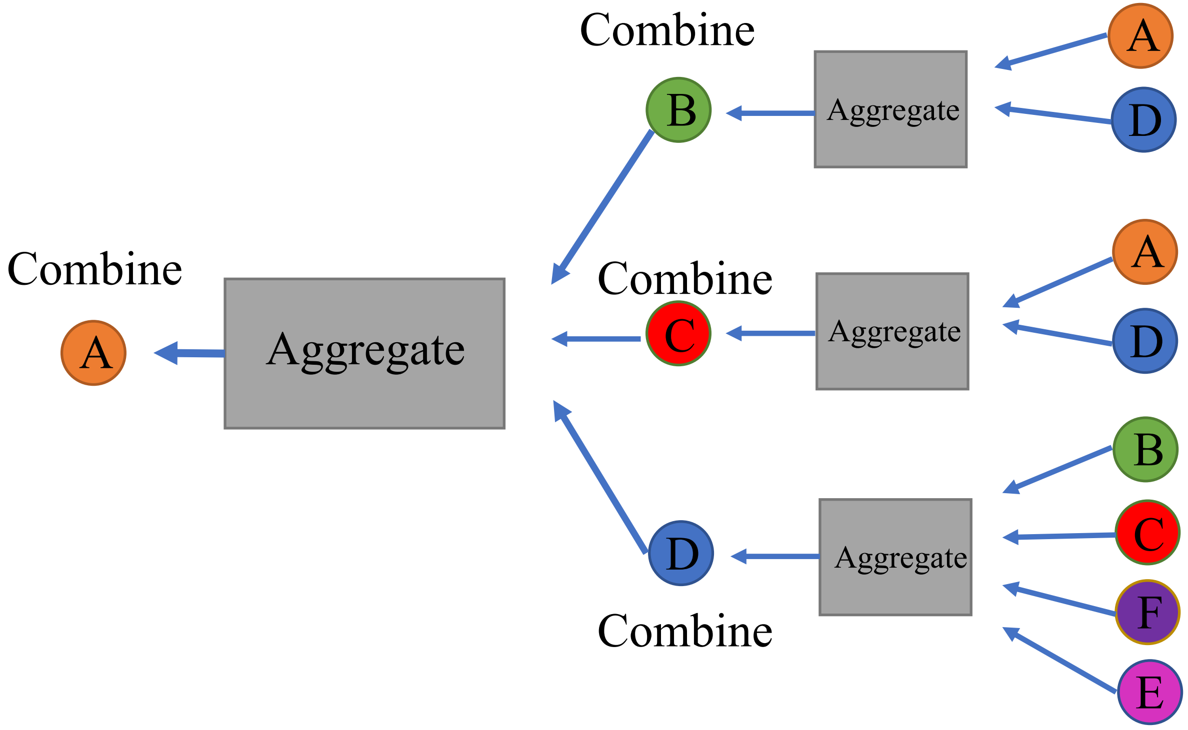


Cryptocurrency Network

Graph Data and Graph Neural Networks

- Generally, Graph Neural Network (GNN) follows the message passing paradigm

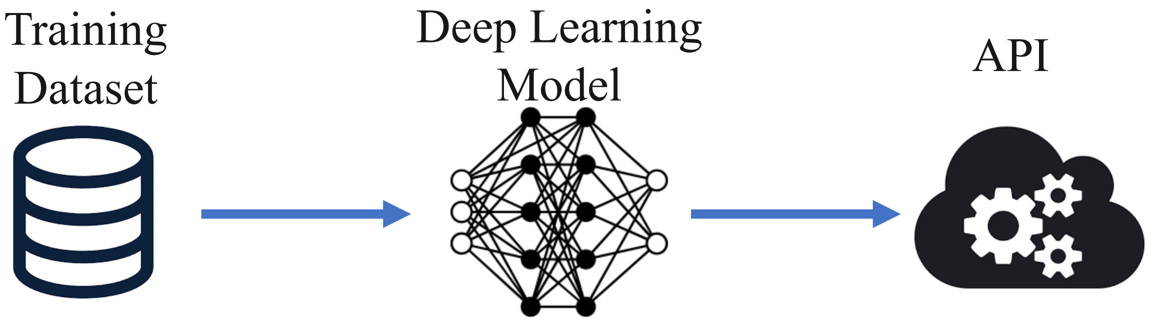




$$h_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, \mathrm{AGGREGATE}^{(k)}\left(\left\{(h_v^{(k-1)}, h_u^{(k-1)}, e_{uv}) : u \in \mathcal{N}(v)\right\}\right)\right)$$

Privacy Attacks

- According to the attacker’s goal, privacy attacks can be categorized into membership inference attack, model extraction attack and model inversion attack



- Membership Inference: Infer whether one piece of data is in the training dataset
- Model Extraction: “replicate” the deep learning model through the API
- Model Inversion: reconstruct the training dataset from the model

Motivation

- The fact that many GNN based applications such as social relationship analysis rely on processing sensitive graph data raises great privacy concerns
- Studying model inversion attack on GNNs helps us understand the vulnerability of GNN models and enable us to avoid privacy risks in advance

Our Work

- We propose **Graph M**odel **I**nvrsion attack (GraphMI) for edge reconstruction
- Based on GraphMI, we investigate the relation between edge influence and model inversion risk
- Experimental results on several public datasets show the effectiveness of GraphMI

One Motivation Scenario

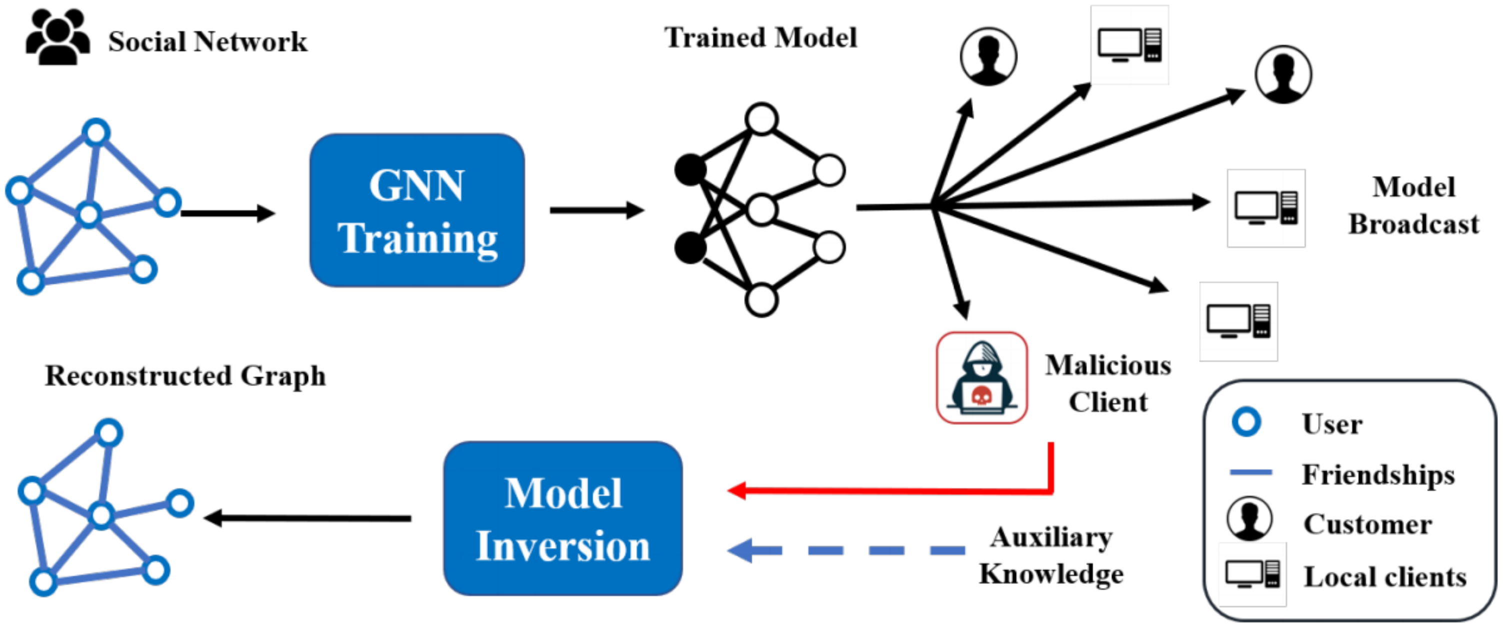


Figure 1: One motivation scenario in social networks

Threat Model

- Attacker's goal
  - The attacker aims to reconstruct the adjacency matrix A of the training graph

- Attacker's Knowledge and Capability
  - White box setting: attacker has access to the target model
  - We assume the attacker has labels of all the node

Model Inversion of Graph Neural Networks

- Let θ be the model parameter of the target model f . During the training phase, f is trained to minimize the loss $\mathcal{L}(\theta, X, A, Y)$

$$\theta^* = \arg\ \min_{\theta} \mathcal{L}(\theta, X, A, Y)$$

- Given the trained model and its parameters, graph model inversion aims to find the adjacency matrix

$$A^* = \arg\ \max_{A} P(A|X, Y, \theta^*)$$

Overview of GraphMI

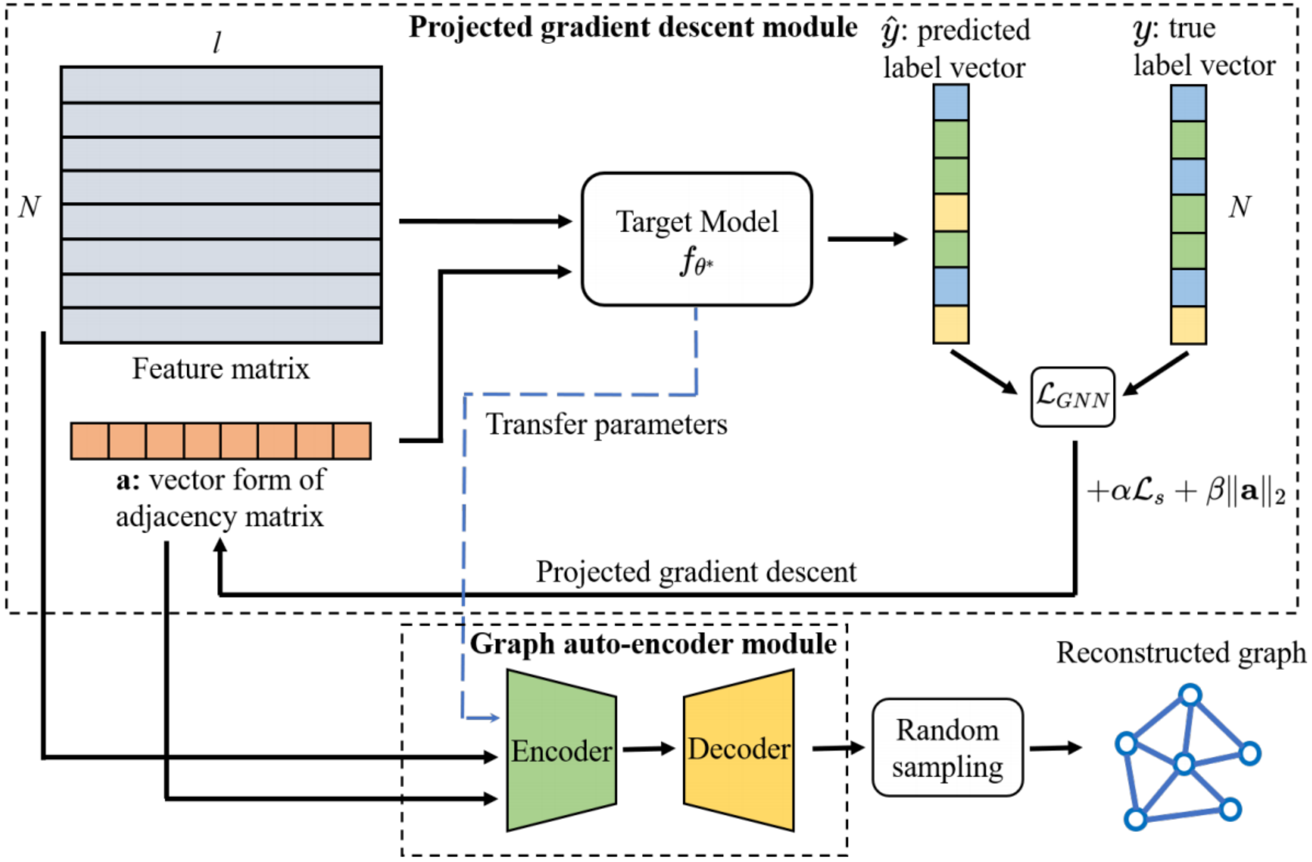


Figure 2: Overview of GraphMI

Proposed Algorithm

- Projected gradient descent module
  - Intuition: the reconstructed adjacency matrix will be similar to the original adjacency matrix if the loss between true labels and predicted labels is minimized

$$\min_{A \in \{0,1\}^{N \times N}} \mathcal{L}_{GNN}(A) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(A, f_{\theta^*}, \mathbf{x}_i, y_i)$$
$$s.t. \quad A = A^\top.$$

- Feature smoothness

$$\mathcal{L}_s = tr(X^\top \hat{L} X) = \frac{1}{2} \sum_{i,j=1}^{N} A_{i,j} \left(\frac{\mathbf{x}_i}{\sqrt{d_i}} - \frac{\mathbf{x}_j}{\sqrt{d_j}}\right)^2$$

Proposed Algorithm

- Final objective function
  - For ease of gradient computation and update, we replace the symmetric reconstructed adjacency matrix A with its vector form and relax it to $\mathbf{a} \in [0, 1]^n$

$$\arg \min_{\mathbf{a} \in [0,1]^n} \mathcal{L}_{attack} = \mathcal{L}_{GNN} + \alpha \mathcal{L}_s + \beta \|\mathbf{a}\|_2.$$

- Projected gradient descent update

$$\mathbf{a}^{t+1} = P_{[0,1]} \left[\mathbf{a}^t - \eta_t g_t \right]$$

$$P_{[0,1]}[x] = \begin{cases} 0 & x < 0 \\ 1 & x > 1 \\ x & otherwise \end{cases}$$

Proposed Algorithm

- Graph Auto-encoder Module

$$A = \mathrm{sigmoid}(ZZ^\top), \text{with } Z = H_{\theta^*}(\mathbf{a}, X)$$

- Random Sampling Module
  - After solving the optimization problem, A can be interpreted as a probabilistic matrix, which represents the possibility of each edge
  - We could use random sampling to recover the binary adjacency matrix

Experimental Settings

- Datasets:

	Nodes	Edges	Classes	Features
Cora	2,708	5,429	7	1,433
Citeseer	3,327	4,732	6	3,703
Polblogs	1,490	19,025	2	-
USA	1,190	13,599	4	-
Brazil	131	1,038	4	-
AIDS	31,385	64,780	38	4
ENZYMES	19,580	74,564	3	18

Table 3: Dataset statistics

- Evaluation Metrics
  - To evaluate our attack, we use AUC (area under the ROC curve) and AP (average precision) as our metrics, which is consistent with previous works

Experimental Results

Method	Cora		Citeseer		Polblogs		USA		Brazil		AIDS		ENZYMES	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Attr. Sim.	0.803	0.808	**0.889**	**0.891**	-	-	-	-	-	-	0.731	0.727	0.564	0.567
MAP	0.747	0.708	0.693	0.755	0.688	0.751	0.594	0.601	0.638	0.661	0.642	0.653	0.617	0.643
GraphMI	**0.868**	**0.883**	0.878	0.885	**0.793**	**0.797**	**0.806**	**0.813**	**0.866**	**0.888**	**0.802**	**0.809**	**0.678**	**0.684**

Table 1: Results of model inversion attack on Graph Neural Networks

- GraphMI achieves the best performance across nearly all the datasets
- One exception is Citesser, which could be explained by more abundant node attribute information of Citeseer compared with other datasets.

Experimental Results

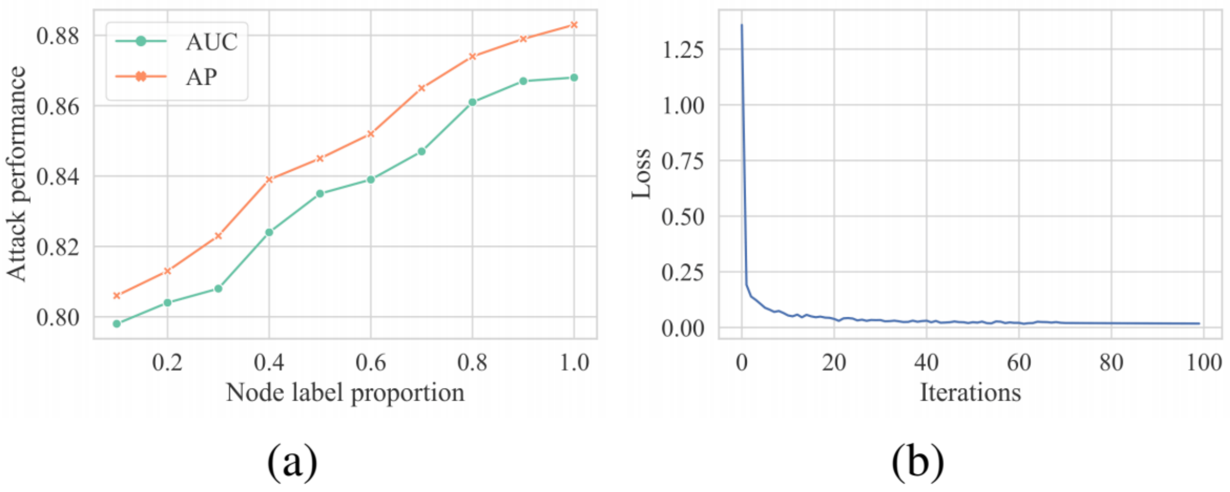


Figure 4: (a) Impact of node label proportion. (b) Convergence plot.

- With fewer node labels, the attack performance will drop
- The loss converges gracefully against iterations, which again verifies the effectiveness of GraphMI

Experimental Results

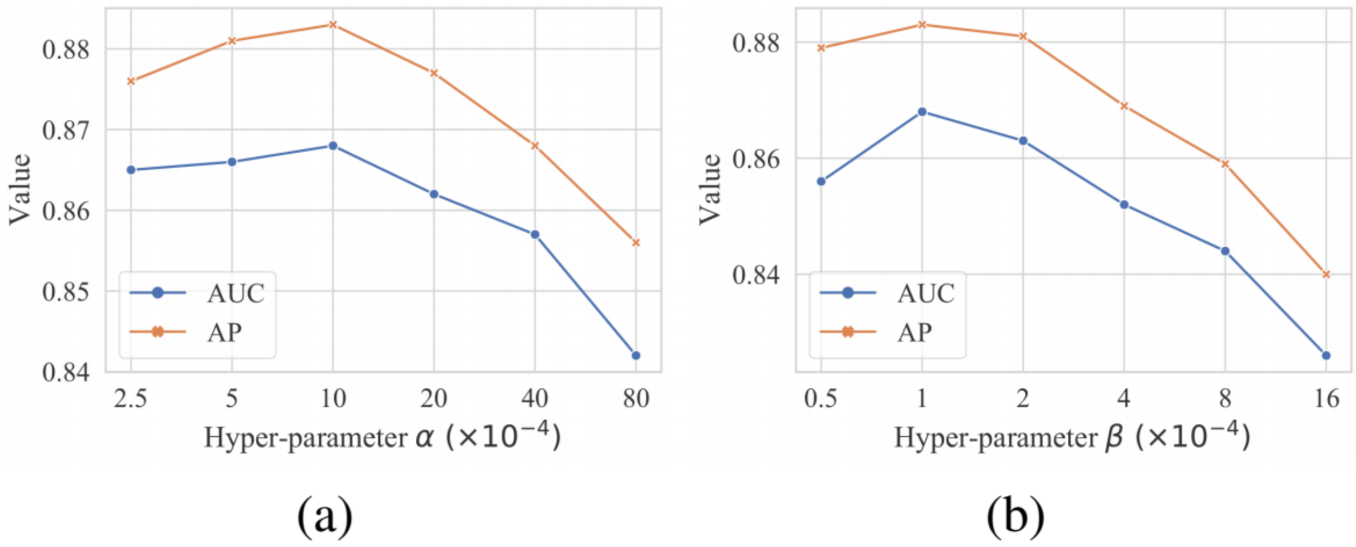


Figure 5: Results of parameter analysis on Cora dataset

- The attack performance of GraphMI can be boosted when choosing proper values for all the hyper-parameters

Experimental Results

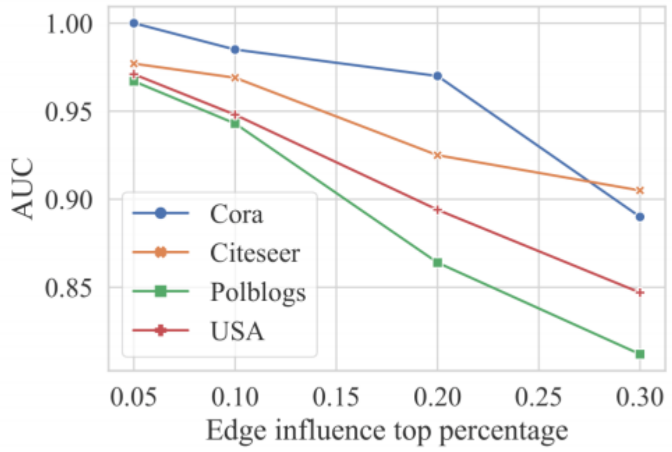


Figure 6: Impact of edge influence on the performance of the GraphMI attack.

$$\mathcal{I}(e) = ACC(f_{\theta^*}, A, X) - ACC(f_{\theta^*}, A_{-e}, X)$$

$$ACC(f_{\theta^*}, A, X) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(f^i_{\theta^*}(A, X) = y_i)$$

- Edges with greater influence are more likely to be inferred successfully through model inversion attack.

Experimental Results

Method	ACC	GraphMI AUC
$\epsilon = 1.0$	0.48	0.60
$\epsilon = 5.0$	0.65	0.72
$\epsilon = 10.0$	0.78	0.84
no DP	0.80	0.87

Table 2: The performance of the GraphMI attack against GCN trained with differential privacy on Cora dataset

- As the privacy budget ϵ drops, the performance of GraphMI attack deteriorates at the price of a huge utility drop.
- Generally, enforcing DP on target models cannot prevent GraphMI attack

Conclusion

- In this paper, we presented GraphMI, a model inversion attack method against Graph Neural Networks
- Extensive experimental results showed its effectiveness on several state-of-the-art graph neural networks.
- We also explored and evaluated the impact of node label proportion, edge influence and differential privacy on the attack performance

- Future Works:
  - Extend the current work to a black-box setting
  - Design countermeasures with a better trade-off between utility and privacy

Thank you!

- For any further questions, please email :

  zaixi@mail.ustc.edu.cn