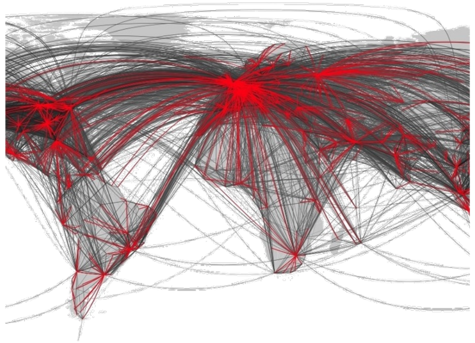
Jointly Attacking Graph Neural Network and its Explanations

Wenqi Fan[1*], Han Xu[2*], Wei Jin[2], Xiaorui Liu[3], Xianfeng Tang[4], Suhang Wang[5], Qing Li[1], Jiliang Tang[2], Jianping Wang[6], and Charu Aggarwal[7]

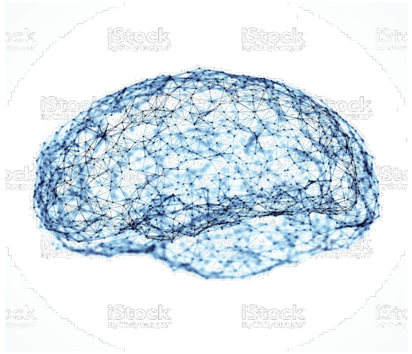
[1]The Hong Kong Polytechnic University, [2]Michigan State University, [3]North Carolina State University, [4]Amazon, [5]The Pennsylvania State University, [6]City University of Hong Kong, [7]IBM T.J. Watson

Data as Graphs

Social Graphs

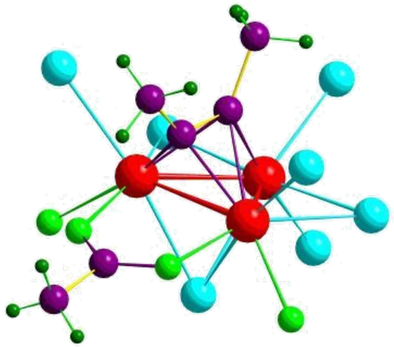


Transportation Graphs

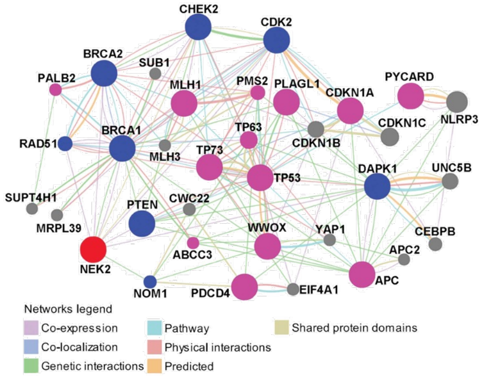


Brain Graphs

Web Graphs



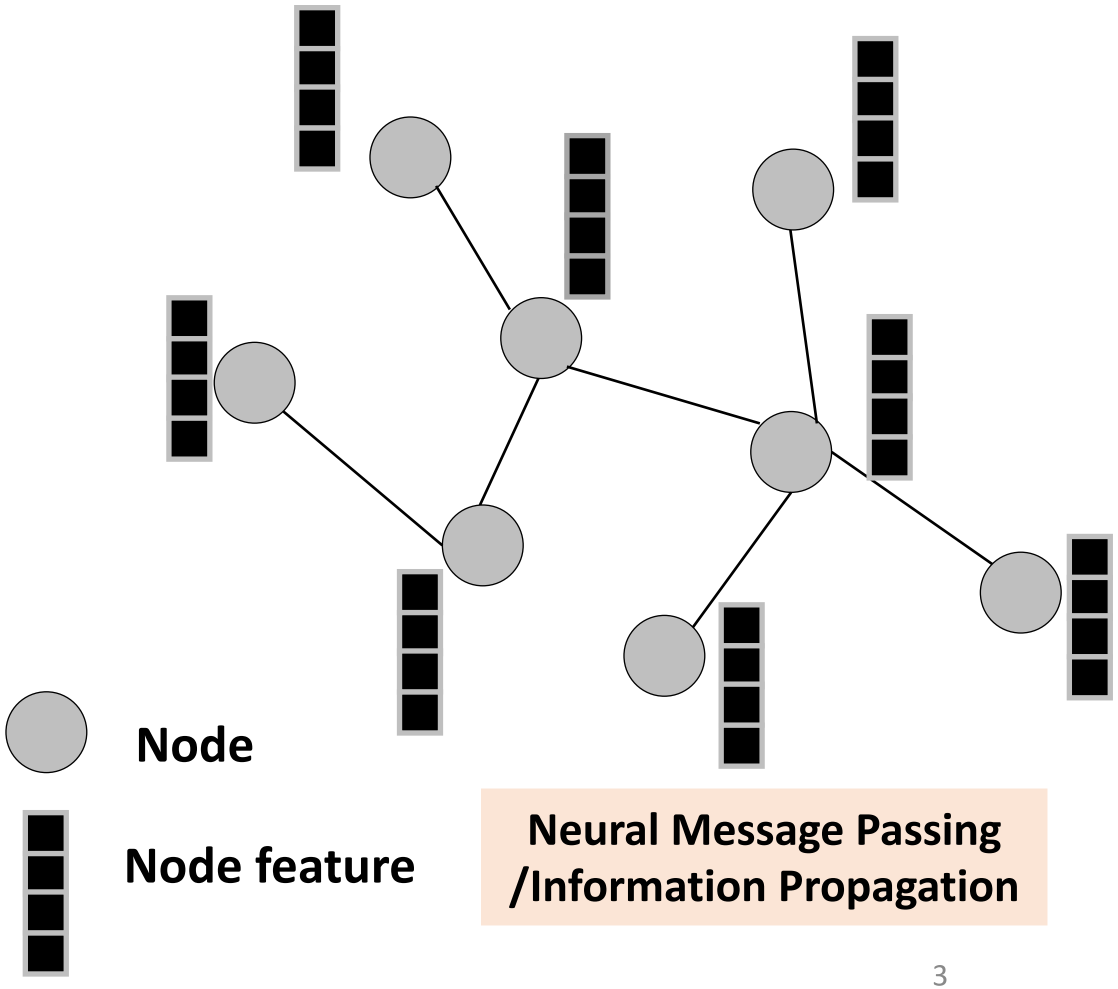
Molecular Graphs



Gene Graphs

Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods [Message Passing].

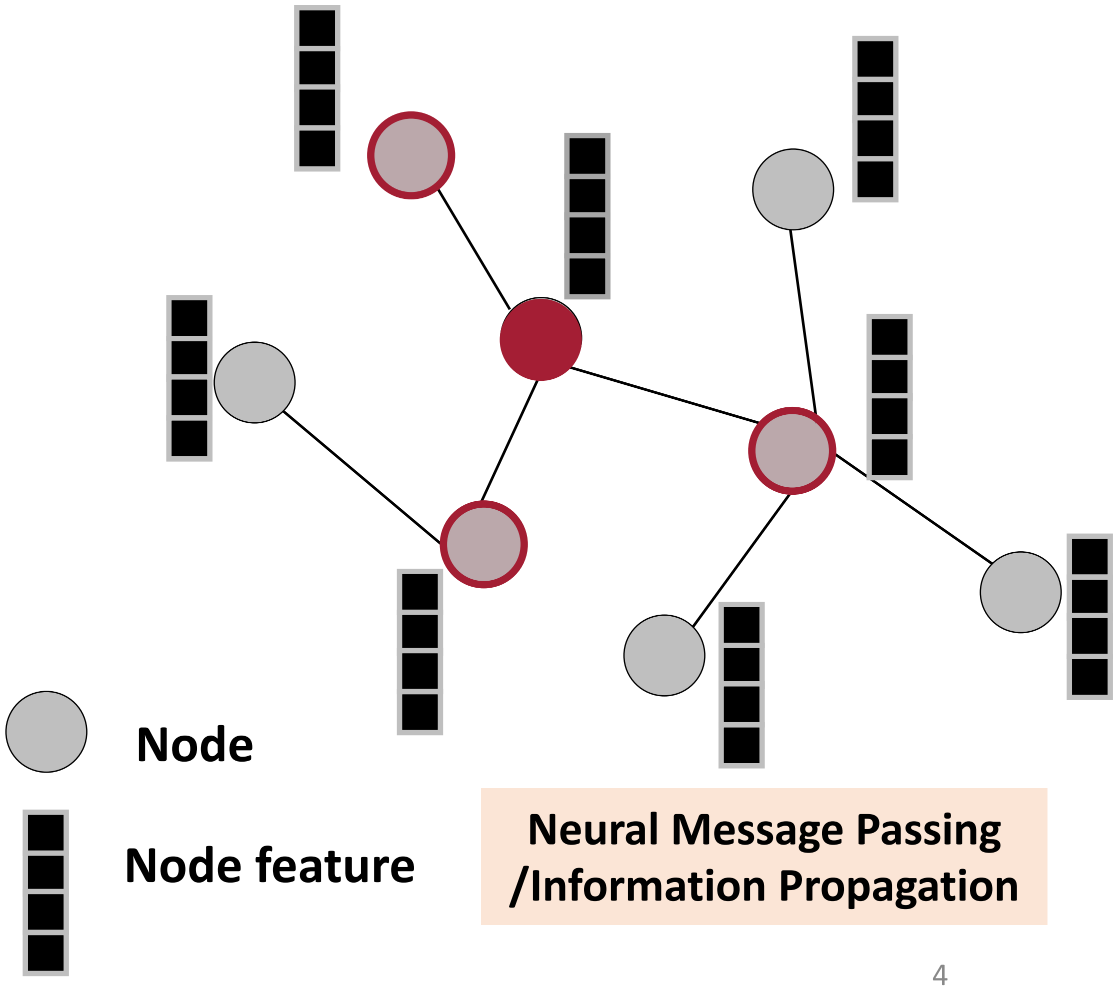


Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods [Message Passing].

Model a local structural information (neighborhood) of a node;



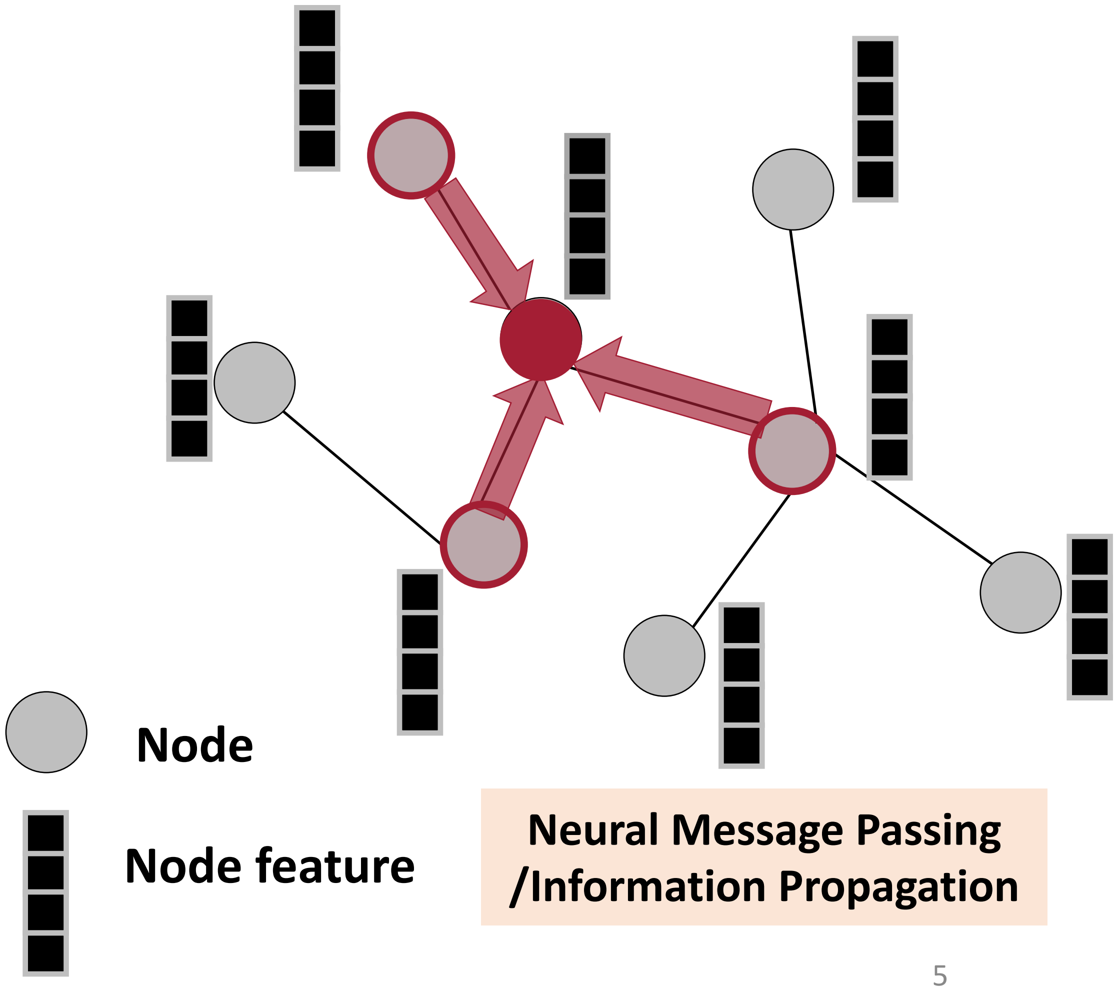
Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods [Message Passing].

1. Model a local structural information (neighborhood) of a node;
2. Aggregation operation
3. Representation update

GNNs can naturally integrate node feature and the topological structure for graph-structured data.

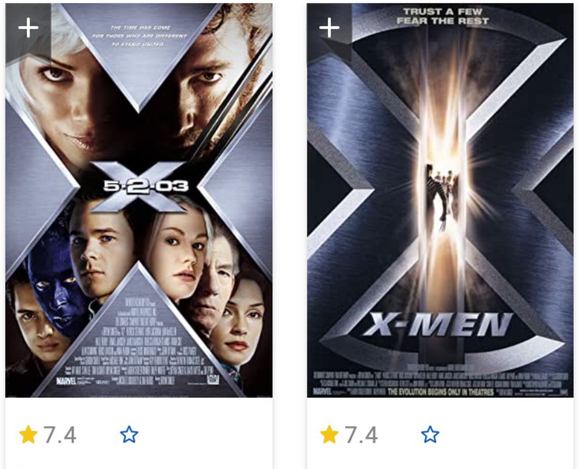


Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

GNNs-based System is Everywhere

Business

Healthcare



Entertainment

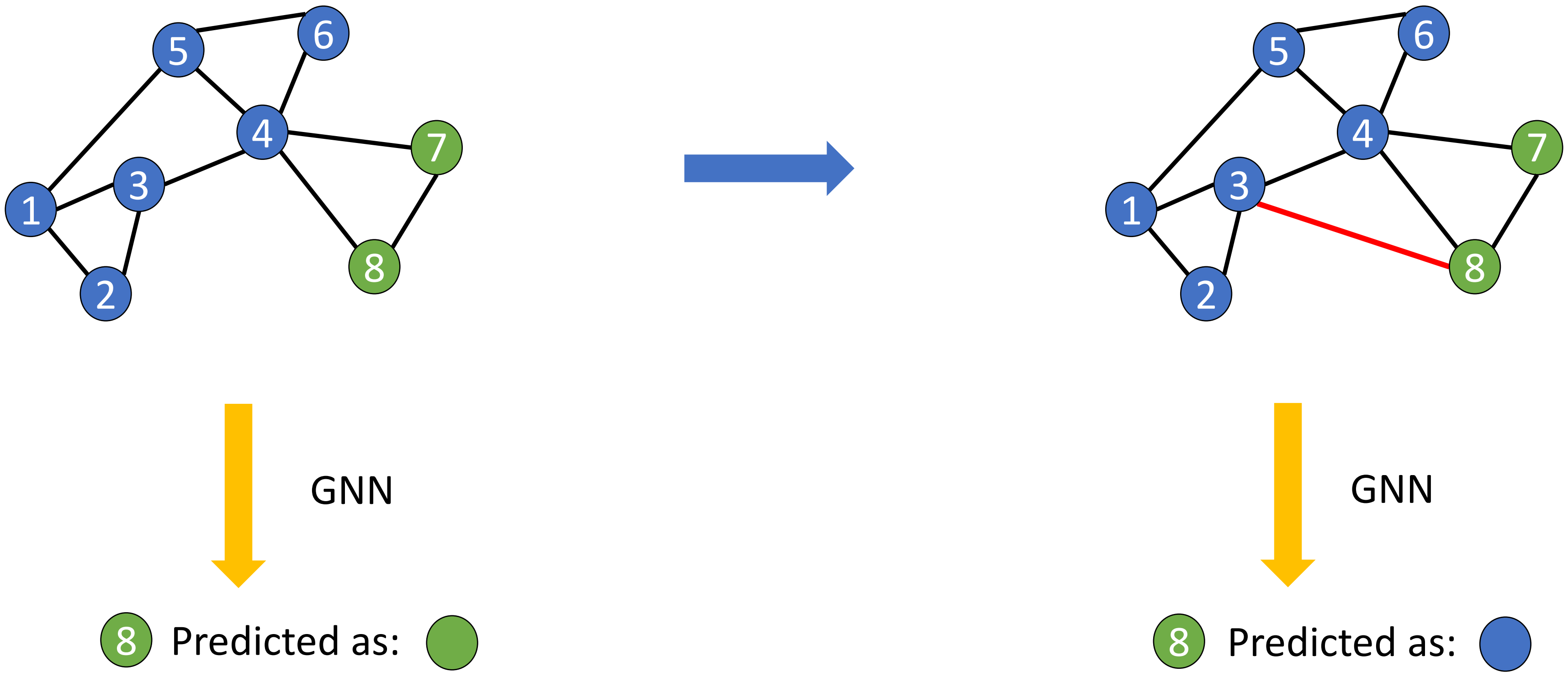
Education

Adversarial Attacks on Deep Learning

Classified as panda	Small adversarial noise	Classified as gibbon
x	ϵ	x'

Find x' satisfying $||x' - x|| \leq \Delta$
such that $C(x') \neq y$

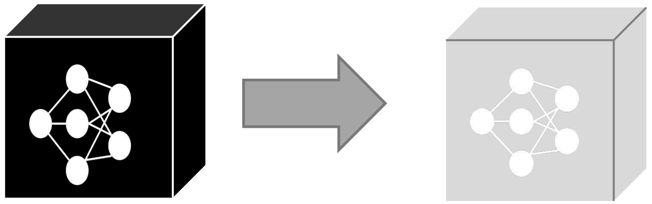
Adversarial Attacks on GNNs



GNNs Explainability

How GNNs make decision?

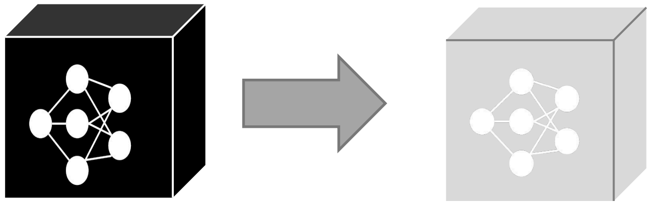
From Black-box to
“Transparent”

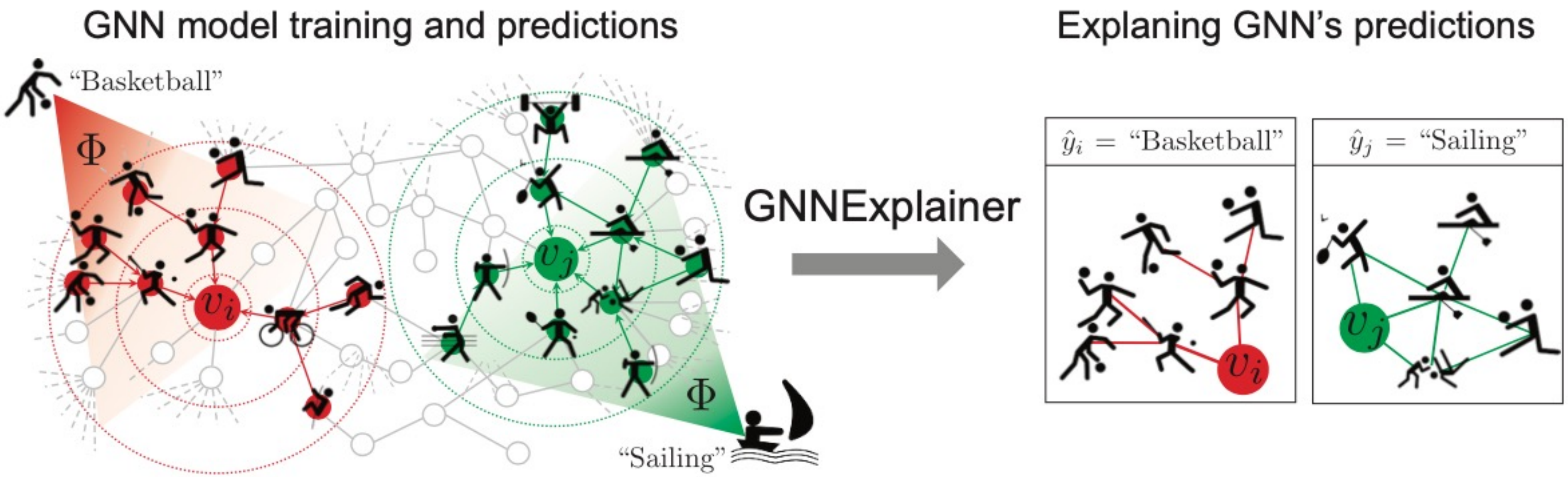


GNNs Explainability

How GNNs make decision?

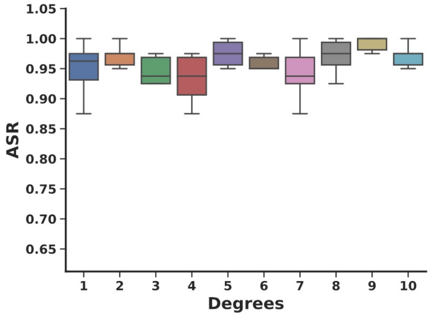
From Black-box to
"Transparent"



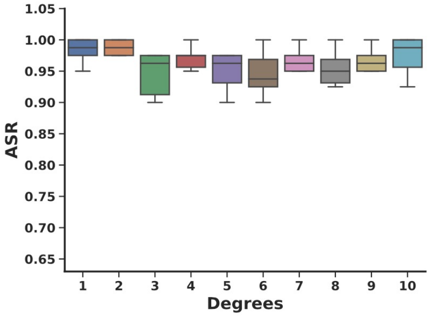


GNNExplainer as Adversarial Inspector

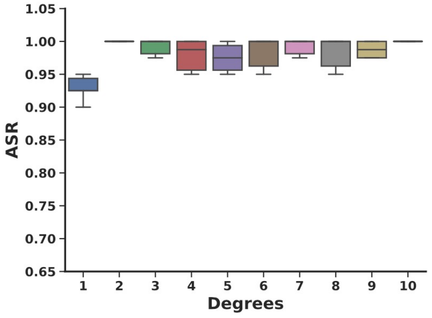
GNNExplainer can act as an inspection tool and have the potential to detect the adversarial perturbations for graphs.



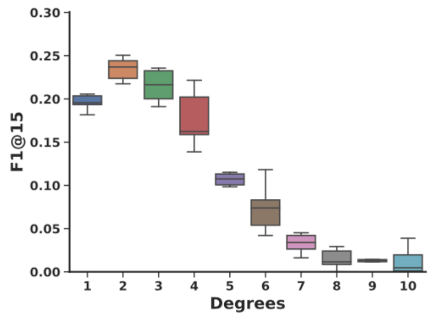
(a) CITESEER



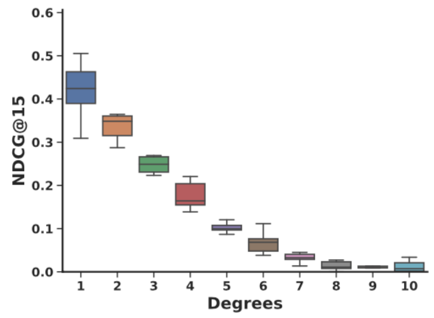
(b) CORA



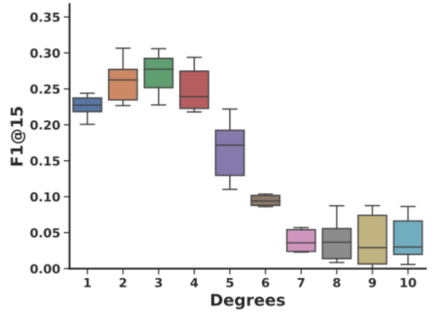
(c) ACM



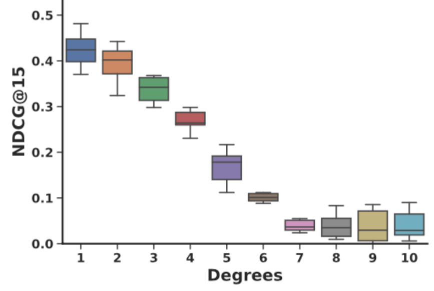
(a) CITESEER - F1@15



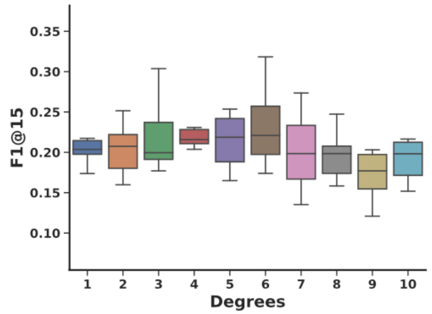
(b) CITESEER - NDCG@15



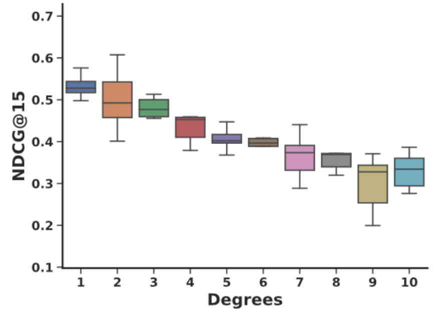
(c) CORA - F1@15



(d) CORA - NDCG@15



(e) ACM - F1@15

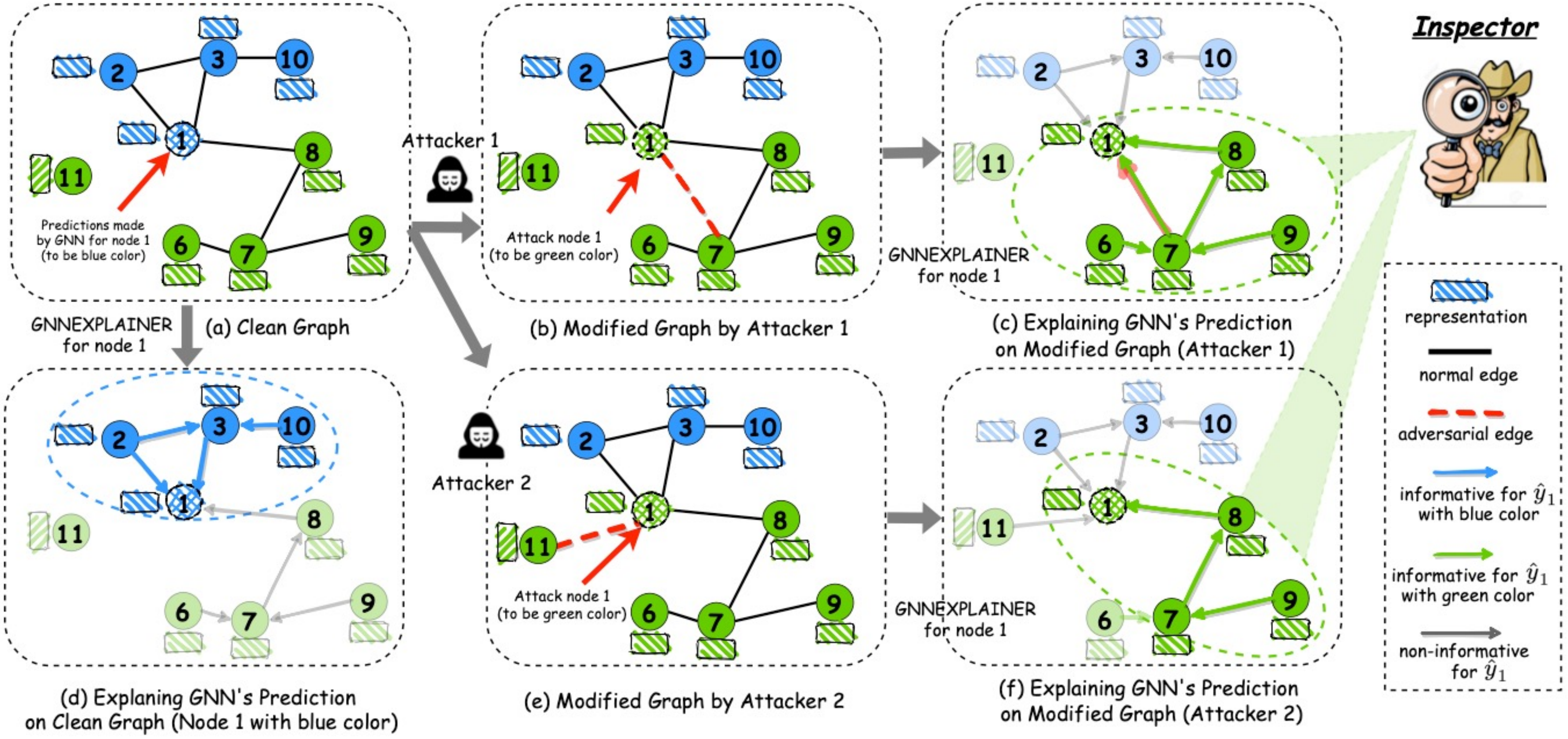


(f) ACM - NDCG@15

Research Problem

Whether a graph neural network and its explanations can be jointly attacked by modifying graphs with malicious desires?

Research Problem



Adversarial attacks and the explanations for prediction made by a GNN model.

Problem Statement

Problem: Given $G = (\mathbf{A}, \mathbf{X})$, target (victim) nodes $v_i \subseteq V_t$ and specific target label $\hat{y}_i$, the attacker aims to select adversarial edges to composite a new graph $\hat{\mathbf{A}}$ which fulfills the following two goals:

- The added adversarial edges can change the GNN's prediction to a specific target label: $\hat{y}_i = \arg\max_c f_\theta(\hat{\mathbf{A}}, \mathbf{X})^c_{v_i}$;
- The added adversarial edges will not be included in the subgraph generated by GNNEXPLAINER: $\hat{\mathbf{A}} - \mathbf{A} \notin \mathbf{A}_S$.

Formulation

Node Classification

Two-layer GCN model

$$f_\theta(\mathbf{A},\mathbf{X}) = \mathrm{softmax}(\tilde{\mathbf{A}}\,\sigma(\tilde{\mathbf{A}}\,\mathbf{X}\,\mathbf{W}_1)\,\mathbf{W}_2)$$

$$\begin{aligned}\min_\theta\ \mathcal{L}_{\text{GNN}}(f_\theta(\mathbf{A},\mathbf{X})) &:= \sum_{v_i\in V_L} \ell\left(f_\theta(\mathbf{A},\mathbf{X})_{v_i}, y_i\right) \qquad (1)\\ &= -\sum_{v_i\in V_L}\sum_{c=1}^{C}\mathbb{I}[y_i=c]\ln(f_\theta(\hat{\mathbf{A}},\mathbf{X})_{v_i}^c)\end{aligned}$$

GNNExplainer

$$\begin{aligned}&\max_{(\mathbf{A}_S,\mathbf{X}_S)} MI\,(Y,(\mathbf{A}_S,\mathbf{X}_S))\\ \to\ &\min_{(\mathbf{A}_S,\mathbf{X}_S)} H(Y|\mathbf{A}=\mathbf{A}_S,\mathbf{X}=\mathbf{X}_S)\\ \approx\ &\min_{(\mathbf{A}_S,\mathbf{X}_S)} -\sum_{c=1}^{C}\mathbb{I}[\hat{y}_i=c]\ln f_\theta(\mathbf{A}_S,\mathbf{X}_S)_{v_i}^c\end{aligned}$$

Adversarial Edges

$$\begin{aligned}&\min_{\mathbf{M}_A}\mathcal{L}_{\text{Explainer}}(f_\theta,\mathbf{A},\mathbf{M}_A,\mathbf{X},v_i,\hat{y}_i)\\ &\to\max_{\mathbf{M}_A}\sum_{c=1}^{C}\mathbb{I}[\hat{y}_i=c]\ln f_\theta(\mathbf{A}\odot\sigma(\mathbf{M}_A),\mathbf{X})_{v_i}^c\end{aligned}$$

Graph Attack

$$\min_{\hat{\mathbf{A}}} \mathcal{L}_{\text{GNN}}(f_\theta(\hat{\mathbf{A}}, \mathbf{X})_{v_i}, \hat{y}_i) := -\sum_{c=1}^{C} \mathbb{I}[\hat{y}_i = c] \ln(f_\theta(\hat{\mathbf{A}}, \mathbf{X})_{v_i}^{c})$$

Perturbation budget: $\|\mathbf{E}'\| = \|\hat{\mathbf{A}} - \mathbf{A}\|_0 \leq \Delta.$

- **Gradient-based attack methods**

  Discrete property in Graph -> Relax the adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ as continuous variable.

GNNExplainer Attack

$$\min_{\hat{\mathbf{A}}} \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{M}_A^T[i,j] \cdot \mathbf{B}[i,j] \tag{9}$$

where $\mathbf{B} = \mathbf{1}\mathbf{1}^T - \mathbf{I} - \mathbf{A}$. $\mathbf{I}$ is an identity matrix, and $\mathbf{1}\mathbf{1}^T$ is all-ones matrix. $\mathbf{1}\mathbf{1}^T - \mathbf{I}$ corresponds to the fully-connected graph. When t is 0, $\mathbf{M}_A^0$ is randomly initialized; while t is larger than 0, $\mathbf{M}_A^t$ is updated as follows:

$$\mathbf{M}_A^t = \mathbf{M}_A^{t-1} - \eta \nabla_{\mathbf{M}_A^{t-1}} \boxed{\mathcal{L}_{\text{Explainer}}(f_\theta, \hat{\mathbf{A}}, \mathbf{M}_A^{t-1}, \mathbf{X}, v_i, \hat{y}_i)}.$$

$$\rightarrow \max_{\mathbf{M}_A} \sum_{c=1}^{C} \mathbb{I}[\hat{y}_i = c] \ln f_\theta(\mathbf{A} \odot \sigma(\mathbf{M}_A), \mathbf{X})_{v_i}^c$$

Sophisticated dependency

$$\mathbf{M}_A^0 \rightarrow \mathbf{M}_A^1 \rightarrow \cdots \rightarrow \mathbf{M}_A^T$$

Our Proposed GEAttack

Bi-level optimization problem:

$$\min_{\hat{\mathbf{A}}} \mathcal{L}_{\text{GEAttack}} := \mathcal{L}_{\text{GNN}}(f_\theta(\hat{\mathbf{A}}, \mathbf{X})_{v_i}, \hat{y}_i) + \lambda \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{M}_A^T[i,j] \cdot \mathbf{B}[i,j].$$

where $\mathbf{M}_A^0$ is randomly initialized when t is 0, and for $t > 0$, $\mathbf{M}_A^t$ can be updated as follows:

$$\mathbf{M}_A^t = \mathbf{M}_{A^{t-1}} - \eta \nabla_{\mathbf{M}_A^{t-1}} \mathcal{L}_{\text{Explainer}}(f_\theta, \hat{\mathbf{A}}, \mathbf{M}_A^{t-1}, \mathbf{X}, v_i, \hat{y}_i).$$

Inner Loop

- Mimic the optimization process of GNNExperliner
- Maintain the computation graph of these updates on dependency of adjacency mask matrix

Outer Loop

- Require high-order gradient computation by the Automatic Differentiation Package

Our Proposed GEAttack

Algorithm 1 GEAttack

Input: perturbation budget: Δ; step-size and update iterations of GNNExplainer: η, T; target node v_i; target label $\hat{y}_i$; graph $G = (\mathbf{A}, \mathbf{X})$, and a GNN model: f_θ.

Output: the adversarial adjacency matrix $\hat{\mathbf{A}}$.

$\mathbf{B} = \mathbf{1}\mathbf{1}^T - \mathbf{I} - \mathbf{A}$, $\hat{\mathbf{A}} = \mathbf{A}$, and randomly initialize $\mathbf{M}_A^0$;

for $o = 1, 2, \ldots, \Delta$ **do** // outer loop over $\hat{\mathbf{A}}$;

  for $t = 1, 2, \ldots, T$ **do** // inner loop over $\mathbf{M}_A^t$;

    compute $\mathbf{P}^t = \nabla_{\mathbf{M}_A^{t-1}} \mathcal{L}_{\text{Explainer}}(f_\theta, \hat{\mathbf{A}}, \mathbf{M}_A^{t-1}, \mathbf{X}, v_i, \hat{y}_i)$;

    gradient descent: $\mathbf{M}_A^t = \mathbf{M}_A^{t-1} - \eta\, \mathbf{P}^t$;

  end for

  compute the gradient w.r.t. $\hat{\mathbf{A}}$: $\mathbf{Q}^o = \nabla_{\hat{\mathbf{A}}} \mathcal{L}_{\text{GEAttack}}$;

  select the edge between node pair (v_i, v_j) with the maximum element $\mathbf{Q}^o[i, j]$ as the adversarial edge, and update $\hat{\mathbf{A}}[i, j] = 1$ and $\mathbf{B}[i, j] = 0$;

end for

Return $\hat{\mathbf{A}}$.

Experiment

Table 1: Results with standard deviations (±std) on three datasets using different attacking algorithms.

Dataset	Metrics (%)	FGA[3]	RNA	FGA-T	Nettack	IG-Attack	FGA-T&E	GEAttack
CITERSEER	**ASR**	86.79±0.08	55.52±0.08	99.56±0.01	99.11±0.01	91.54±0.05	98.74±0.02	**100±0.00**
	ASR-T	-	54.27±0.10	99.56±0.01	99.11±0.01	91.54±0.05	98.74±0.02	**100±0.00**
	Precision	13.45±0.01	9.96±0.01	13.44±0.02	10.21±0.01	10.21±0.01	13.31±0.01	**9.87±0.02**
	Recall	74.55±0.05	**63.80±0.05**	74.55±0.05	66.48±0.06	65.73±0.04	74.28±0.05	64.05±0.07
	F1	21.65±0.02	**16.44±0.02**	21.64±0.02	17.08±0.02	16.96±0.02	21.47±0.02	16.49±0.03
	NDCG	47.18±0.04	39.21±0.04	46.60±0.04	38.45±0.05	40.26±0.04	47.02±0.05	**36.11±0.05**
CORA	**ASR**	90.54±0.05	62.97±0.10	**100±0.00**	**100±0.00**	90.17±0.07	99.79±0.01	**100±0.00**
	ASR-T	-	62.58±0.10	**100±0.00**	**100±0.00**	90.17±0.07	99.79±0.01	**100±0.00**
	Precision	16.02±0.01	**10.47±0.01**	16.08±0.01	12.78±0.01	13.47±0.03	15.95±0.01	12.21±0.01
	Recall	72.65±0.05	**55.40±0.07**	72.75±0.05	63.83±0.06	67.66±0.04	72.45±0.05	65.03±0.06
	F1	25.30±0.02	**17.00±0.02**	25.38±0.02	20.64±0.02	21.79±0.04	25.21±0.02	20.06±0.02
	NDCG	43.15±0.04	**34.16±0.05**	43.41±0.04	36.47±0.04	38.05±0.05	43.46±0.04	35.60±0.03
ACM	**ASR**	67.50±0.07	63.66±0.13	**100±0.00**	98.00±0.03	98.82±0.02	**100±0.00**	**100±0.00**
	ASR-T	-	63.66±0.13	**100±0.00**	98.00±0.03	98.82±0.02	**100±0.00**	**100±0.00**
	Precision	11.57±0.05	**9.26±0.01**	11.88±0.05	12.98±0.03	11.69±0.05	11.31±0.05	9.61±0.02
	Recall	38.21±0.12	**34.05±0.05**	38.34±0.12	43.67±0.09	44.49±0.14	37.90±0.12	38.08±0.08
	F1	14.16±0.05	**12.75±0.02**	14.35±0.05	17.61±0.04	16.61±0.07	13.91±0.05	14.03±0.03
	NDCG	38.58±0.14	36.68±0.10	38.17±0.13	46.90±0.09	41.23±0.13	38.07±0.13	**24.43±0.06**

[3] FGA cannot evaluate ASR-T metric where the specific target label are not available.

- **GEAttack works consistently comparable to or outperform other strong GNN attacking methods.**
- **GEAttack consistently outperforms other methods when attacking the GNNExplainer, except for the RNA method.**
- **Both GNNs model and its explanations are vulnerable to adversarial attacks**

Conclusion

- GNNExplainer (as Adversarial Inspector) can be utilized to understand and inspect the problematic outputs from adversarially perturbed graph data.
- **A new attacking problem:** jointly attack a graph neural network method and its explanations.
- Our proposed algorithm GEAttack successfully resolves the dilemma between attacking GNN and its explanations by exploiting their vulnerabilities simultaneously.
- The very first study: investigate **interactions** between adversarial attacks and explainability for the trustworthy GNNs.

ICDE 2023
Anaheim, CA April 3 – 7, 2023

THANK YOU

Jointly Attacking Graph Neural Network and its Explanations

wenqi.fan@polyu.edu.hk / xuhan1@msu.edu