When Witnesses Defend: A Witness Graph Topological Layer for Adversarial Graph Learning.

Naheed Anjum Arafat, Debabrota Basu, Yulia Gel, Yuzhou Chen

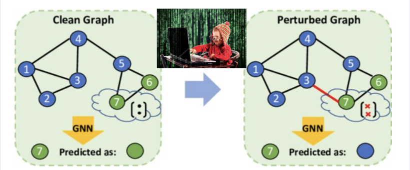
Post-doc
Nanyang Technological University, Singapore (2021-2024)
PhD
National University of Singapore (2020)

February 11, 2025

Adversarial attack: a Modern Challenge to GNNs

Adversarial attack on Graph learning algorithms.

Attacker misleads a learning algorithm (e.g. GNN) into making incorrect predictions or classifications by deliberately perturbing a small number of edges (e.g. remove/add edges) or node features.



Adversarial perturbation (around target node 7) causes misclassification.

Contributions

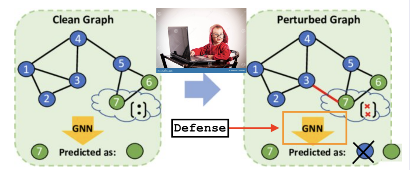
1. We introduced a novel topological adversarial defense, namely, the *Witness Graph Topological Layer (WGTL)*.
2. WGTL integrates local and global higher-order graph characteristics and controls their potential defense role via a topological regularizer.

Adversarial attack: a Modern Challenge to GNNs

Adversarial attack on Graph learning algorithms.

Attacker misleads a learning algorithm (e.g. GNN) into making incorrect predictions or classifications by deliberately perturbing a small number of edges (e.g. remove/add edges) or node features.

Problem

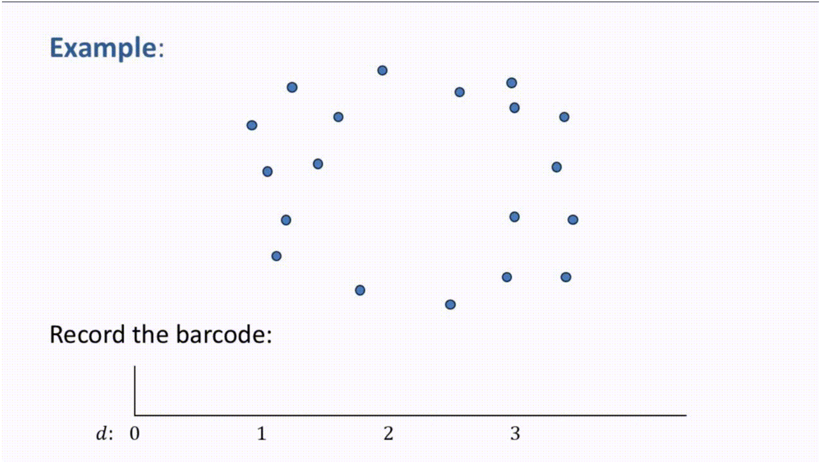


Design a defense algorithm that mitigates the effect of adversarial attack

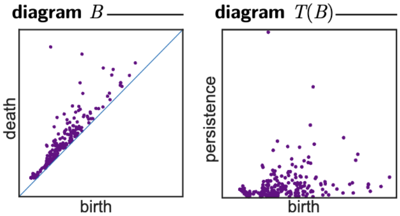
Contributions

1. We introduced a novel topological adversarial defense, namely, the *Witness Graph Topological Layer (WGTL)*.
2. WGTL integrates local and global higher-order graph characteristics and controls their potential defense role via a topological regularizer.

Topological Features



Topological Features

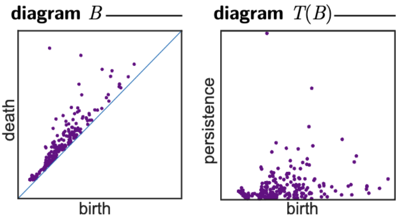


A persistence diagram is transformed using function $T : (x, y) \rightarrow (0, y - x)$.

Why Topological features?

Stability theorem: Small change in the data (graph) only result in small changes in the persistence diagram.

Topological Features



A persistence diagram is transformed using function $T : (x, y) \rightarrow (0, y - x)$.

Why Topological features?

Stability theorem: Small change in the data (graph) only result in small changes in the persistence diagram.

Impact of this work

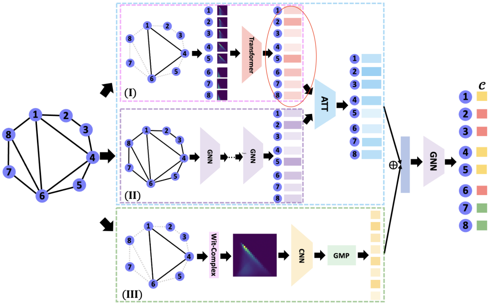
1. This is the first work that shows that topological features can make GNNs robust against adversarial perturbations.
2. Effective against a wide variety of attacks, for instance,
   - Targetted poisoning attack (Graybox, modify the neighbors of a target node and their features)
   - Global poisoning attack (Graybox, instead of targetting specific neighborhood modify whichever edges minimizes the model accuracy)
   - Adaptive attacks (White-box, the model architecture, parameters and defense mechanisms are known to the attacker)
   - Node feature attack
3. WGTL improves existing defenses such as Pro-GNN, GNNGuard, and SimP-GCN respectively by 5%, 15%, and 5%.

Impact of this work

1. This is the first work that shows that topological features can make GNNs robust against adversarial perturbations.
2. Effective against a wide variety of attacks, for instance,
   - Targetted poisoning attack (Graybox, modify the neighbors of a target node and their features)
   - Global poisoning attack (Graybox, instead of targetting specific neighborhood modify whichever edges minimizes the model accuracy)
   - Adaptive attacks (White-box, the model architecture, parameters and defense mechanisms are known to the attacker)
   - Node feature attack
3. WGTL improves existing defenses such as Pro-GNN, GNNGuard, and SimP-GCN respectively by 5%, 15%, and 5%.

Impact of this work

1. This is the first work that shows that topological features can make GNNs robust against adversarial perturbations.
2. Effective against a wide variety of attacks, for instance,
   - Targetted poisoning attack (Graybox, modify the neighbors of a target node and their features)
   - Global poisoning attack (Graybox, instead of targetting specific neighborhood modify whichever edges minimizes the model accuracy)
   - Adaptive attacks (White-box, the model architecture, parameters and defense mechanisms are known to the attacker)
   - Node feature attack
3. WGTL improves existing defenses such as Pro-GNN, GNNGuard, and SimP-GCN respectively by 5%, 15%, and 5%.



Architecture of Witness Graph Topological Layer.

1. **Local Topology Encoding:** Encodes local topological features of every node. (Z_{T_L})
2. Node Representation Learning. Learns node representations using any backbone GNN. (Z_G)
3. Global Topology Encoding. Encodes topological feature of the entire graph. (Z_{T_G})
4. Aggregated Topological Encoding. Encodes local and global topological priors. (Z_{WGTL})

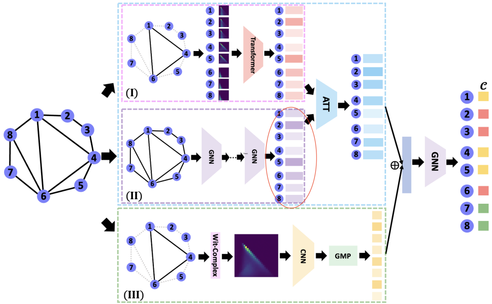
$$z = [Z_{T_L}, Z_G]$$

$$\text{Attention coefficients, } \alpha_i = \mathbf{Softmax}(\boldsymbol{W_2} \cdot \tanh(\boldsymbol{W_1} z_i + \boldsymbol{b_1}))$$

$$\text{Additive attention, } Z_{\mathrm{AGG}} = \alpha_1 \times Z_{T_L} + \alpha_2 \times Z_G$$

$$Z_{\mathrm{WGTL}} = Z_{\mathrm{AGG}} Z_{T_G}$$

WGTL: Topological Encodings



Architecture of Witness Graph Topological Layer.

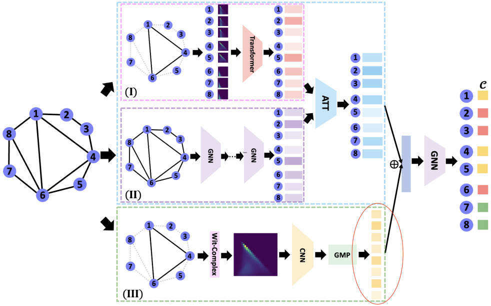
1. **Local Topology Encoding:** Encodes local topological features of every node. (Z_{T_L})
2. **Node Representation Learning.** Learns node representations using any backbone GNN. (Z_G)
3. Global Topology Encoding. Encodes topological feature of the entire graph. (Z_{T_G})
4. Aggregated Topological Encoding. Encodes local and global topological priors. (Z_{WGTL})

$$z = [Z_{T_L}, Z_G]$$

$$\text{Attention coefficients, } \alpha_i = \mathbf{Softmax}(\boldsymbol{W_2} \cdot \tanh(\boldsymbol{W_1} z_i + \boldsymbol{b_1}))$$

$$\text{Additive attention, } Z_{\mathrm{AGG}} = \alpha_1 \times Z_{T_L} + \alpha_2 \times Z_G$$

$$Z_{\mathrm{WGTL}} = Z_{\mathrm{AGG}} Z_{T_G}$$



Architecture of Witness Graph Topological Layer.

1. **Local Topology Encoding:** Encodes local topological features of every node. (Z_{T_L})
2. **Node Representation Learning.** Learns node representations using any backbone GNN. (Z_G)
3. **Global Topology Encoding.** Encodes topological feature of the entire graph. (Z_{T_G})
4. **Aggregated Topological Encoding.** Encodes local and global topological priors. (Z_{WGTL})

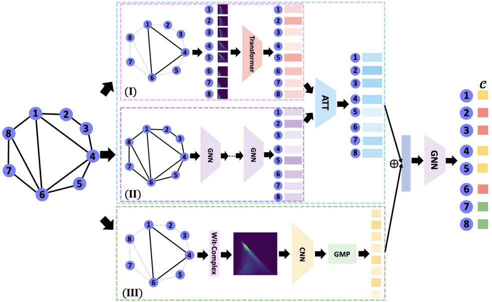
$$z = [Z_{T_L}, Z_G]$$

$$\text{Attention coefficients, } \alpha_i = \textbf{Softmax}(\boldsymbol{W_2} \cdot \tanh(\boldsymbol{W_1} z_i + \boldsymbol{b_1}))$$

$$\text{Additive attention, } Z_{\mathrm{AGG}} = \alpha_1 \times Z_{T_L} + \alpha_2 \times Z_G$$

$$Z_{\mathrm{WGTL}} = Z_{\mathrm{AGG}} Z_{T_G}$$

WGTL: Topological Encodings



Architecture of Witness Graph Topological Layer.

1. **Local Topology Encoding:** Encodes local topological features of every node. (Z_{T_L})
2. **Node Representation Learning**. Learns node representations using any backbone GNN. (Z_G)
3. **Global Topology Encoding.** Encodes topological feature of the entire graph. (Z_{T_G})
4. **Aggregated Topological Encoding.** Encodes local and global topological priors. (Z_{WGTL})

$$z = [Z_{T_L}, Z_G]$$

$$\text{Attention coefficients, } \alpha_i = \textbf{Softmax}(\boldsymbol{W_2} \cdot \tanh(\boldsymbol{W_1} z_i + \boldsymbol{b_1}))$$

$$\text{Additive attention, } Z_{\mathrm{AGG}} = \alpha_{\mathbf{1}} \times Z_{T_L} + \alpha_{\mathbf{2}} \times Z_G$$

$$Z_{\mathrm{WGTL}} = Z_{\mathrm{AGG}} Z_{T_G}$$

WGTL: Topological Regularizer

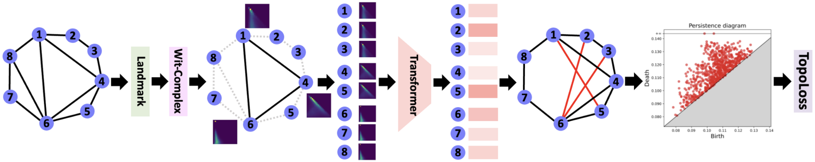


Illustration of Witness Complex-based topological regularizer L_{Topo}.

$$L_{topo}(\mathrm{T}(\mathcal{G})) \triangleq \sum_{i=1}^{m} (d_i - b_i)^2 \left(\frac{d_i + b_i}{2} \right)^2, \tag{1}$$

- A localized attack (perturbing certain nodes or edges) appears as topological noise in the final persistent diagram, and exhibit lower persistence.
- And minimising L_{topo} forces the Transformer to learn local topology encodings (Z_{T_L}) which produces PD with small persistence, i.e., $(d_i - b_i)$.

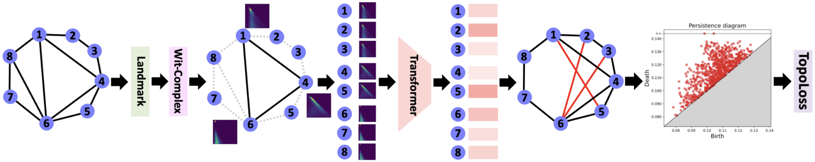


Illustration of Witness Complex-based topological regularizer L_{Topo}.

$$L_{topo}(\mathrm{T}(\mathcal{G})) \triangleq \sum_{i=1}^{m} (d_i - b_i)^2 \left(\frac{d_i + b_i}{2} \right)^2, \qquad (1)$$

- A localized attack (perturbing certain nodes or edges) appears as topological noise in the final persistent diagram, and exhibit lower persistence.
- And minimising L_{topo} forces the Transformer to learn local topology encodings (Z_{T_L}) which produces PD with small persistence, i.e., $(d_i - b_i)$.

Effectiveness

Table 1: Comparison of performances (avg. accuracy±std.) with existing defenses under mettack.

Dataset	Models	Perturbation Rate		
		0%	5%	10%
Cora-ML	Pro-GNN	82.98±0.23	80.14±1.34	71.59±1.33
	Pro-GNN+WGTL	**83.85±0.38**	**81.90±0.73**	**72.51±0.76**
	GCN+GNNGuard	83.21±0.34	76.57±0.50	69.13±0.77
	GCN+GNNGuard+WGTL	*__84.78±0.43__	*__83.23±0.82__	*__79.96±0.49__
	SimP-GCN	79.52±1.81	74.75±1.40	70.87±1.70
	SimP-GCN+WGTL	**81.49±0.52**	**76.65±0.65**	**72.88±0.83**
Citeseer	ProGNN	72.34±0.99	68.96±0.67	67.36±1.12
	ProGNN+WGTL	**72.83±0.94**	**71.85±0.74**	**70.70±0.57**
	GCN+GNNGuard	71.82±0.43	70.79±0.22	66.86±0.54
	GCN+GNNGuard+WGTL	**73.37±0.63**	**72.57±0.17**	**66.93±0.21**
	SimP-GCN	73.73±1.54	73.06±2.09	72.51±1.25
	SimP-GCN+WGTL	*__74.32±0.19__	*__74.05±0.71__	*__73.09±0.50__
Pubmed	Pro-GNN	87.33±0.18	87.25±0.09	87.20±0.12
	Pro-GNN + WGTL (ours)	**87.90±0.30**	*__87.77±0.08__	*__87.67±0.22__
	GCN+GNNGuard	83.63±0.08	79.02±0.14	76.58±0.16
	GCN+GNNGuard+WGTL	OOM	OOM	OOM
	SimP-GCN	*88.11±0.10	86.98±0.19	86.30±0.28
	SimP-GCN+WGTL	OOM	OOM	OOM
Polblogs	GCN+GNNGuard	95.03±0.25	73.25 ±0.16	72.76±0.75
	GCN+GNNGuard+WGTL	*__96.22±0.25__	*__73.62±0.22__	*__73.72±1.00__
	SimP-GCN	89.78±6.47	65.75±5.03	61.53±6.41
	SimP-GCN+WGTL	**94.56±0.24**	**69.78±4.10**	**69.55±4.42**

Efficiency

Table 2: Efficiency of WGTL. All the times are in seconds.

Datasets/ (# Landmarks)	Landmark selection time	Local feat. comput. time	Global feat. comput. time
Cora-ML/124	0.01±0.01	0.12±0.03	5.11±0.13
Citeseer/105	0.01±0.01	0.16±0.02	5.23±1.22
Polblogs/61	0.01±0.00	0.07±0.01	4.64±0.2
Snap-patents/91	0.03±0.02	0.64±0.00	7.54±1.15
Pubmed/394	0.07±0.01	0.51±0.03	27.83±0.47
OGBN-arXiv/84	1.02 ±0.00	12.79±0.31	83.04±2.19