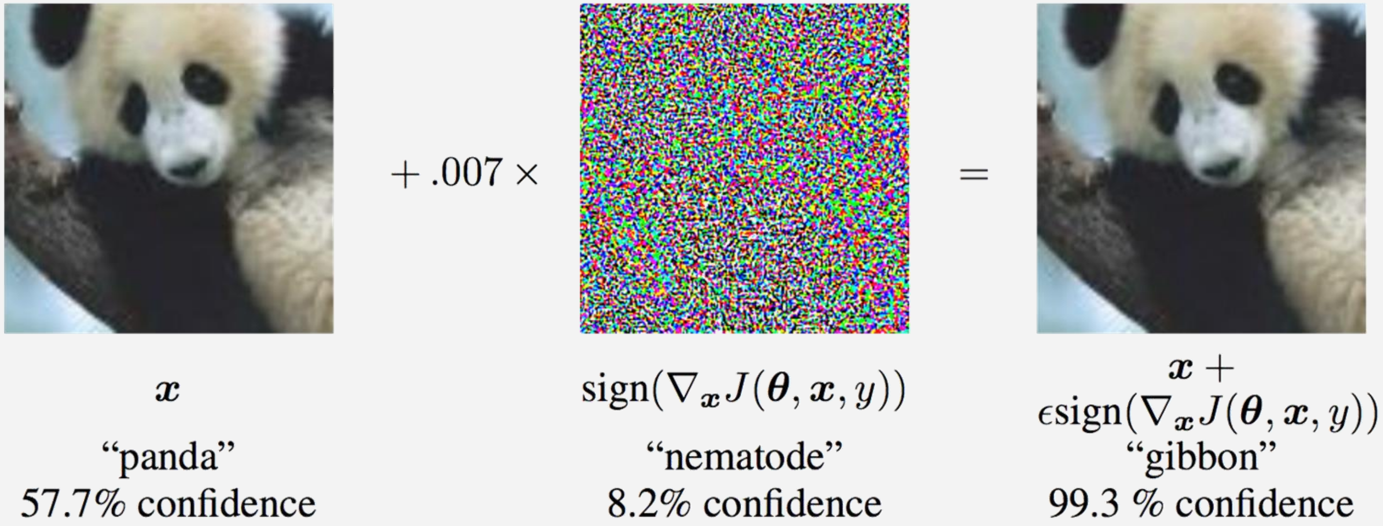
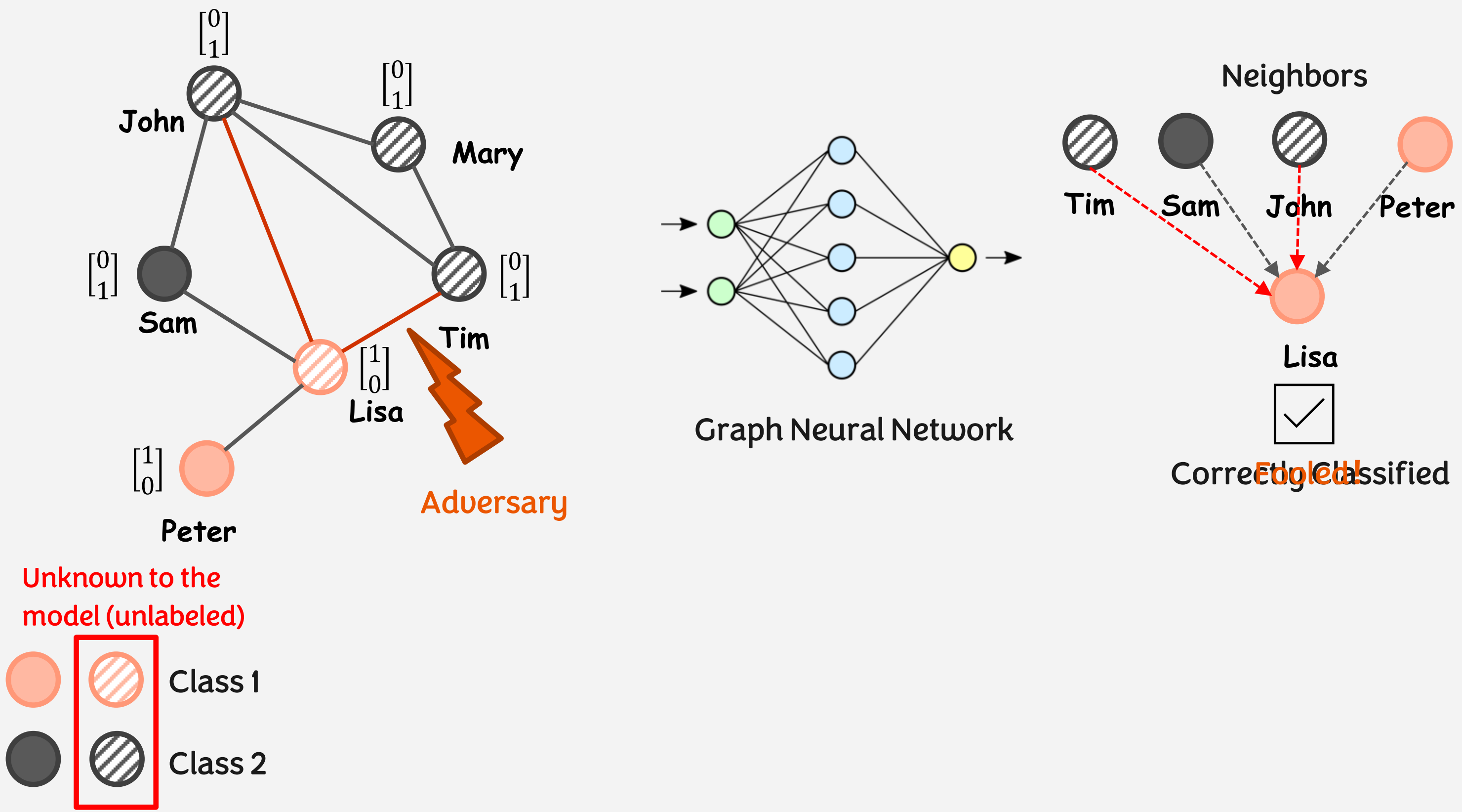
Graph Adversarial Attack

Adversarial Machine Learning

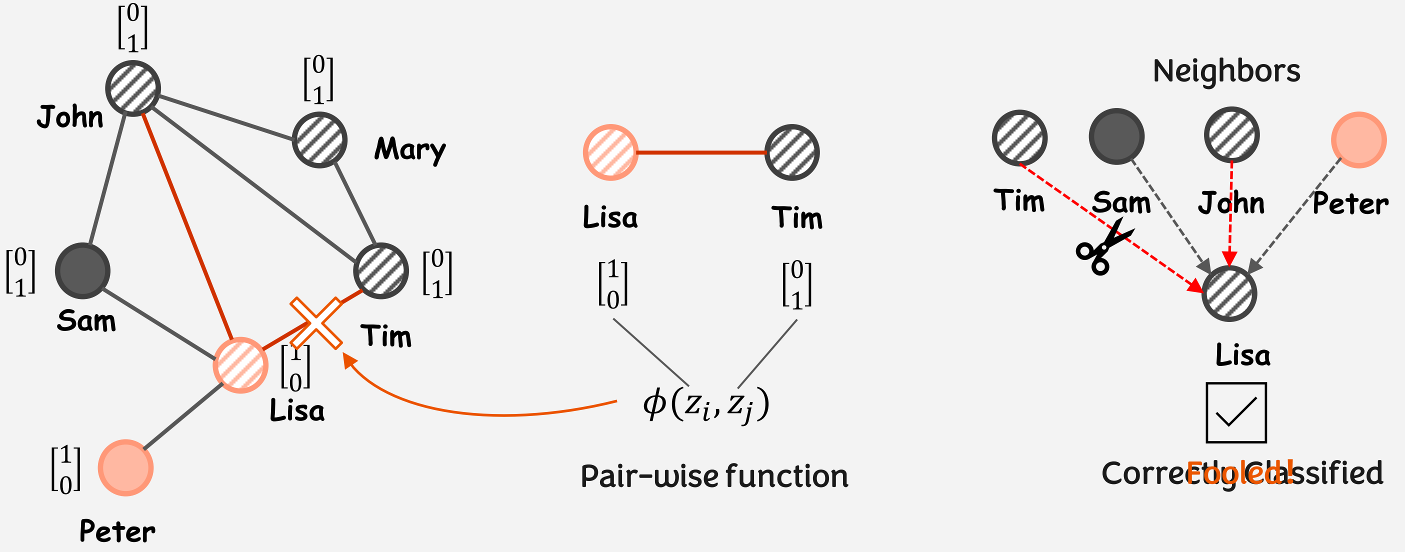


Adversarial Attacks on Graph Structure



Defense: Structure Learning

A straightforward method to deal with the structural perturbation is to find the adversarial edges and remove them.



Background: Existing Methods

Previous Methods

Learn edge weights by a pair-wise metric function --- $\boldsymbol{S}_{ij} = \phi(\mathrm{z_i}, \mathrm{z_j})$, Further, the structure can be optimized according to the weights matrix $\boldsymbol{S}$.

- Compute the function via **original features**: GNNGuard, GCN-Jaccard
- Drawbacks: Lack of structural information – Cause a trade-off.

- Optimize the structure via **representations (task-relevant)** learned by the classifier: GRCN
- Drawbacks: The quality of the representations co-varies with the downstream task performance.

Ptb Rate	GCN	GRCN	GNNGuard	Jaccard
0%	83.56	**86.12**	78.52	81.79
5%	76.36	**80.78**	77.96	80.23
10%	71.62	72.42	**74.86**	74.65
20%	60.31	65.43	72.03	**73.11**

Representations Are The Key

Reliable Representations Make the Defender Stronger:

- Carrying feature information and in the meantime carrying **as much correct structure information** as possible
- **Insensitive** to structural perturbations and **task-irrelevant**

STABLE – an unsupervised pipeline for structure refining

Advantages of Unsupervised Learning

Why is unsupervised learning?

- The unsupervised approach is relatively reliable because the objective is not directly attacked (**task-irrelevant**).

- The unsupervised pipeline can be viewed as a kind of pretraining, and the learned representations may have been trained to be invariant to certain useful properties (**modified structure here**).

Preprocessing and Recovery Schema

We choose graph contrastive learning as our backbone with two robustness-oriented designs

- **Preprocess** the structure by a simple schema: $S_{ij} = sim(x_i, x_j)$

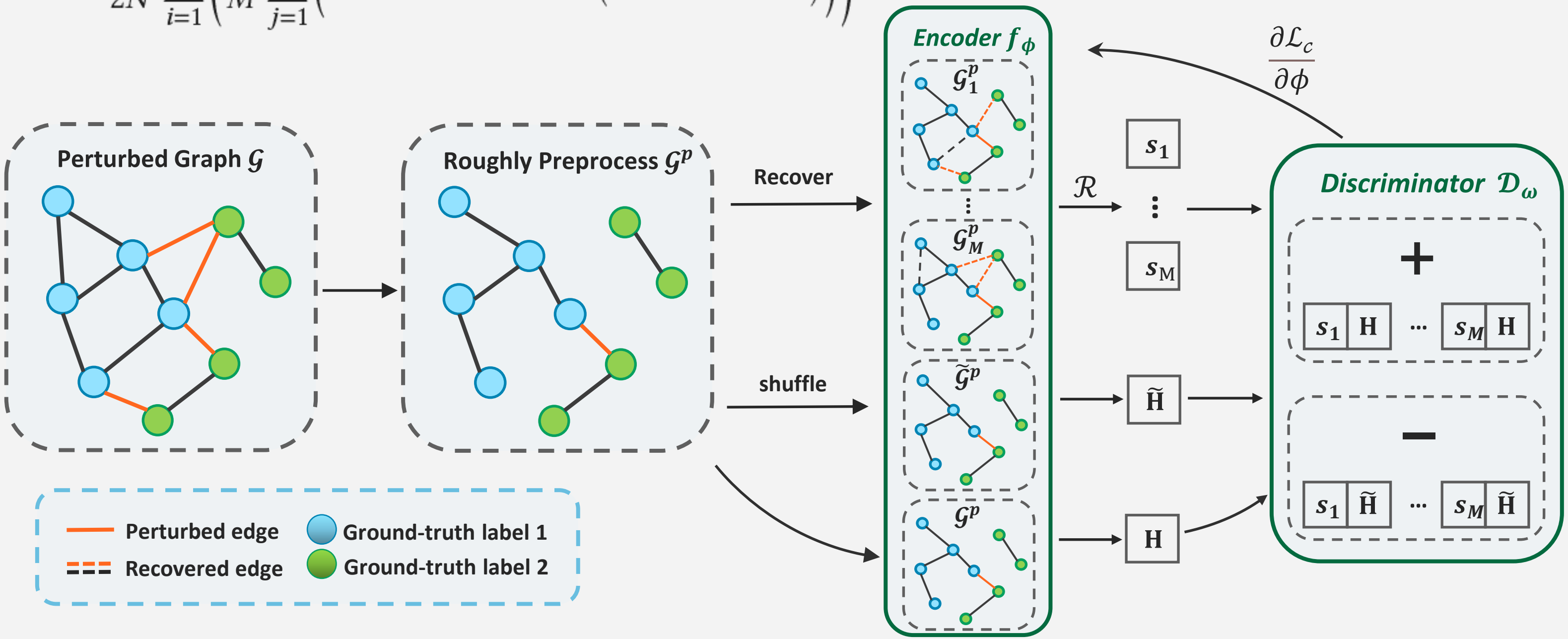
  ——Remove the easily detected adversarial edges

- The augmentation scheme in contrastive methods are naturally similar to adversarial attacks.

  We generate ***M*** views by randomly **recovering** a small portion of the removed edges.

Contrastive Model

$$\mathcal{L}_C = -\frac{1}{2N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{j=1}^{M}\left(\log\mathcal{D}_\omega(\boldsymbol{h}_i, s_j) + \log\left(1-\mathcal{D}_\omega(\tilde{\boldsymbol{h}}_i, s_j)\right)\right)\right).$$



Reliable Representations

Recall our requirements for the reliable representations:

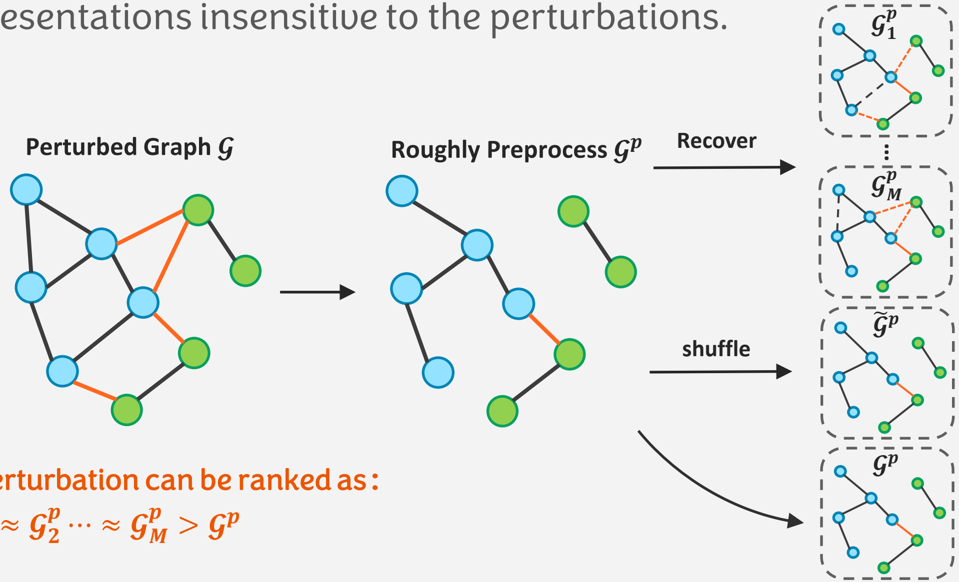
- Carrying feature information and in the meantime carrying **as much correct structure information** as possible

The preprocessing and the effectiveness of contrastive learning meet this requirements.

Reliable Representations

- **Insensitive** to structural perturbations

The recovery can be viewed as injecting slight attacks on $\mathcal{G}^p$, which makes the representations insensitive to the perturbations.



The degrees of perturbation can be ranked as:

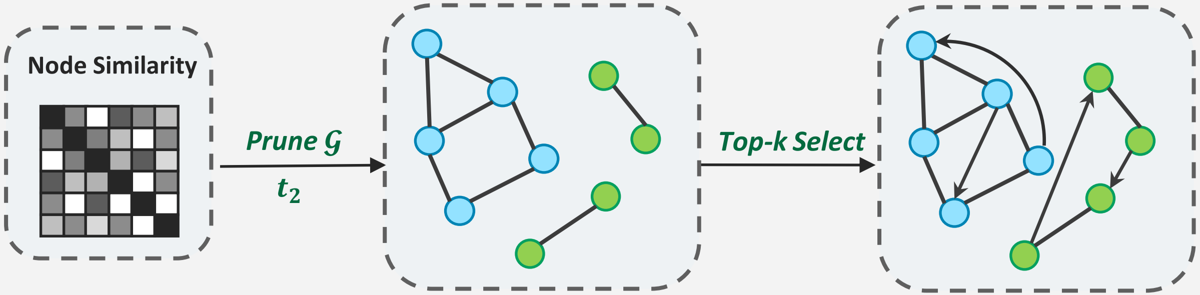
$$\mathcal{G} \gg \mathcal{G}_1^p \approx \mathcal{G}_2^p \cdots \approx \mathcal{G}_M^p > \mathcal{G}^p$$

Graph Refining

We can easily refine the structure by the learned representations.

Prune the graph: $\mathbf{M}_{ij} = \mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j) \longrightarrow \mathbf{A}^R_{ij} = \begin{cases} 1 & \mathit{if}\ \mathbf{M}_{ij} > t_2\ \mathit{and}\ \mathbf{A}_{ij} = 1 \\ 0 & \mathit{otherwise}, \end{cases}$

Add helpful edges --- Link each node with *k* nodes that are most similar to it.



The Vulnerability of GCN

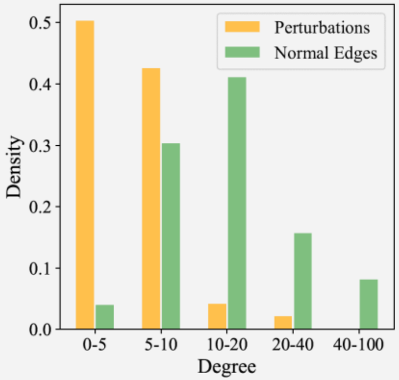
We find GCN suffers from the renormalization trick.

$$\hat{\mathbf{A}} = (\mathbf{D}+\mathbf{I}_N)^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I}_N)(\mathbf{D}+\mathbf{I}_N)^{-\frac{1}{2}}$$

Fake neighbors will be assigned higher weights!

We can trust more on the high-degree neighbors

$$\boldsymbol{h}_i^t = \mathrm{ReLU}\Big(\big(\sum_{j\in\mathcal{N}_i^*}\frac{(d_i d_j)^\alpha}{Z}\boldsymbol{h}_j^{t-1} + \beta\boldsymbol{h}_i^{(t-1)}\big)\mathbf{W}_\theta^t\Big)$$



Attack algorithms tend to link 2 low-degree nodes.

Δ	GCN	GCN*
0%	**83.56**	82.76
5%	76.36	**78.17**
10%	71.62	**74.23**
20%	60.31	**69.59**

Experimental Setup

Datasets

Four public benchmark datasets

- Cora (Citation Graph)
- Citeseer (Citation Graph)
- PubMed (Citation Graph)
- Polblogs (Political Blog Graph)

We only consider the largest connected connected component (LCC).

Datasets	N_{LCC}	E_{LCC}	Classes	Features
Cora	2,485	5,069	7	1433
Citeseer	2,110	3,668	6	3703
Polblogs	1,222	16,714	2	/
PubMed	19717	44338	3	500

Compare methods

Seven robust GNNs under 3 attack methods

- RGCN
- Jaccard
- GNNGuard
- GRCN
- ProGNN
- SimpGCN
- Elastic

- MetaAttack
- DICE
- RANDOM

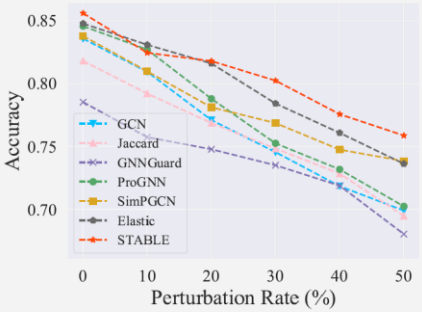
Robustness Evaluation

RQ1: Does STABLE outperform the state-of-the-art defense models under different types of adversarial attacks?

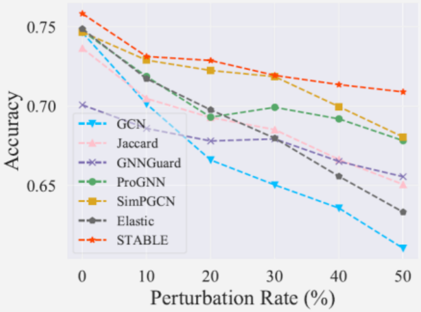
Dataset	Ptb Rate	GCN	RGCN	Jaccard	GNNGuard	GRCN	ProGNN	SimPGCN	Elastic	STABLE
Cora	0%	83.56±0.25	83.85±0.32	81.79±0.37	78.52±0.46	**86.12±0.41**	84.55±0.30	83.77±0.57	84.76±0.53	85.58±0.56
	5%	76.36±0.84	76.54±0.49	80.23±0.74	77.96±0.54	80.78±0.94	79.84±0.49	78.98±1.10	**82.00±0.39**	81.40±0.54
	10%	71.62±1.22	72.11±0.99	74.65±1.48	74.86±0.54	72.43±0.78	74.22±0.31	75.07±2.09	76.18±0.46	**80.49±0.61**
	15%	66.37±1.97	65.52±1.12	74.29±1.11	74.15±1.64	70.72±1.13	72.75±0.74	71.42±3.29	74.41±0.97	**78.55±0.44**
	20%	60.31±1.98	63.23±0.93	73.11±0.88	72.03±1.11	65.34±1.24	64.40±0.59	68.90±3.22	69.64±0.62	**77.80±1.10**
Citeseer	0%	74.63±0.66	75.41±0.20	73.64±0.35	70.07±1.31	75.65±0.21	74.73±0.31	74.66±0.79	74.86±0.53	**75.82±0.41**
	5%	71.13±0.55	72.33±0.47	71.15±0.83	69.43±1.46	**74.47±0.38**	72.88±0.32	73.54±0.92	73.28±0.59	74.08±0.58
	10%	67.49±0.84	69.80±0.54	69.85±0.77	67.89±1.09	72.27±0.69	69.94±0.45	72.03±1.30	73.41±0.36	**73.45±0.40**
	15%	61.59±1.46	62.58±0.69	67.50±0.78	69.14±0.84	67.48±0.42	62.61±0.64	69.82±1.67	67.51±0.45	**73.15±0.53**
	20%	56.26±0.99	57.74±0.79	67.01±1.10	69.20±0.78	63.73±0.82	55.49±1.50	69.59±3.49	65.65±1.95	**72.76±0.53**
Polblogs	0%	95.04±0.11	95.38±0.14	/	/	94.89±0.24	95.93±0.17	94.86±0.46	95.57±0.26	**95.95±0.27**
	5%	77.55±0.77	76.46±0.47	/	/	80.37±0.46	93.48±0.54	75.08±1.08	90.08±1.06	**93.80±0.12**
	10%	70.40±1.13	70.35±0.40	/	/	69.72±1.36	85.81±1.00	68.36±1.88	84.05±1.94	**92.46±0.77**
	15%	68.49±0.49	67.74±0.50	/	/	66.56±0.93	75.60±0.70	65.02±0.74	72.17±0.74	**90.04±0.72**
	20%	68.47±0.54	67.31±0.24	/	/	68.20±0.71	73.66±0.64	64.78±1.33	71.76±0.92	**88.46±0.33**
Pubmed	0%	86.83±0.06	86.02±0.08	86.85±0.09	85.24±0.07	86.72±0.03	87.33±0.18	**88.12±0.17**	87.71±0.06	87.73± 0.11
	5%	83.18±0.06	82.37±0.12	86.22±0.08	84.65±0.09	84.85±0.07	87.25±0.09	86.96±0.18	86.82±0.13	**87.59±0.08**
	10%	81.24±0.17	80.12±0.12	85.64±0.08	84.51±0.06	81.77±0.13	87.25±0.09	86.41±0.34	86.78±0.11	**87.46±0.12**
	15%	78.63±0.10	77.33±0.16	84.57±0.11	84.78±0.10	77.32±0.13	87.20±0.09	85.98±0.30	86.36±0.14	**87.38±0.09**
	20%	77.08±0.2	74.96±0.23	83.67±0.08	84.25±0.07	69.89±0.21	87.09±0.10	85.62±0.40	86.04±0.17	**87.24±0.08**

Robustness Evaluation

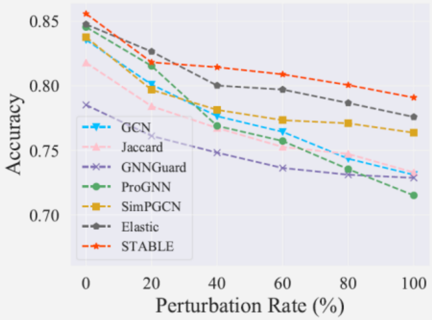
RQ1: Does STABLE outperform the state-of-the-art defense models under different types of adversarial attacks?



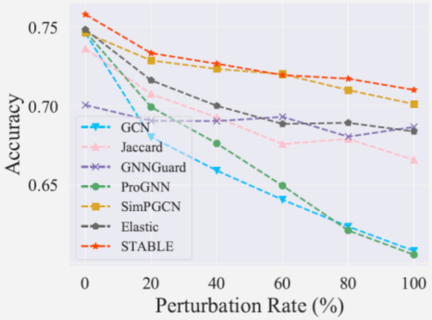
DICE on Cora



DICE on Citeseer



RANDOM on Cora



RANDOM on Citeseer

Result of Sturcture Learning

RQ2: Is the structure learned by STABLE better than learned by other methods?

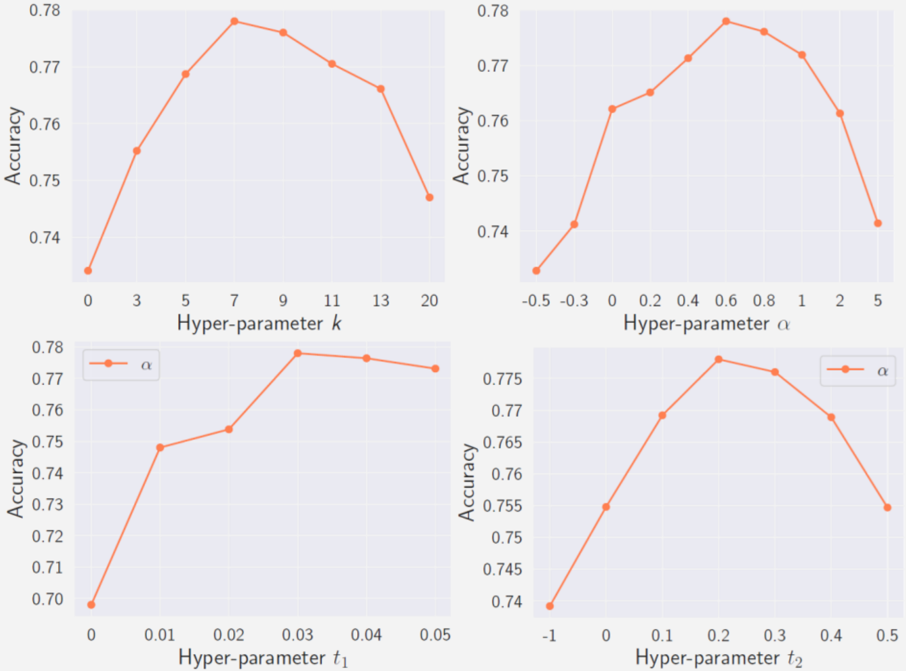
The statistics of the learned graphs

Method	Total	Adversarial	Normal	Accuracy(%)
Jaccard	1,008	447	561	44.35
GNNGuard	1,082	482	600	44.55
STABLE	1,035	601	434	58.07

It can be observed that STABLE achieves the highest pruning accuracy, indicating that STABLE revise the structure more precisely via more reliable representations.

Parameter Analysis

RQ3: What is the performance with respect to different training parameters?

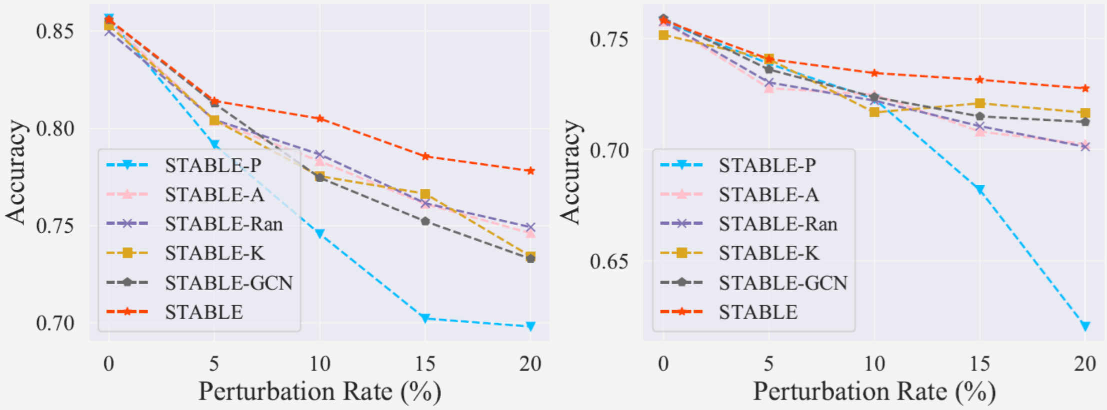


We list the specific values which achieve the best performance on Cora

Ptb Rate	0%	5%	10%	15%	20%	35%	50%
k	1	5	7	7	7	7	13
α	-0.5	-0.3	0.3	0.6	0.6	0.7	0.8

Ablation Study

RQ4: How do the key components benefit the robustness?



(a) Cora (b) Citeseer

Why is Graph Attack so Destructive to GNNs ?

We find a interesting phenomenon which inspires us to revisit this problem from a data distribution perspective.

- We formulate the distribution shift in graph adversarial attack scenario.
- We empirically and theoretically analyze the phenomena in graph attack and defense.
- Then, based on the analysis and observation, we provide nine practical tips to improve existing and future graph attack and defense.

Thanks Q & A

Name: Kuan Li | Email: likuan20s@ict.ac.cn
Homepage: https://likuanppd.github.io/

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES