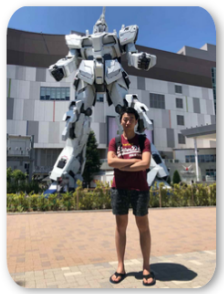
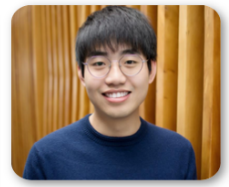
USER: Unsupervised Structural Entropy-based Robust Graph Neural Network

Yifei Wang, Yupan Wang, Zeyu Zhang, Song Yang, Kaiqi Zhao,
Jiamou Liu*

School of Computer Science, The University of Auckland, New Zealand




THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

MARSDEN FUND
TE PŪTEA RANGAHAU A MARSDEN

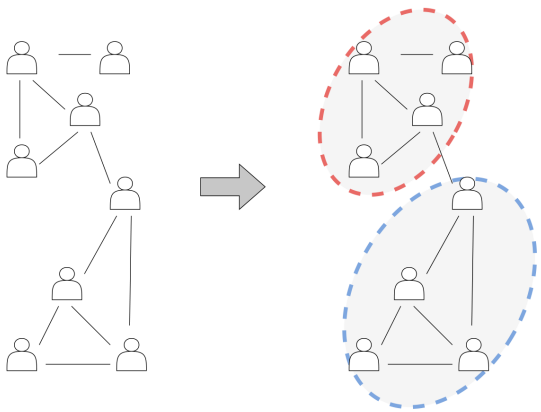
ROYAL SOCIETY TE APĀRANGI

Background

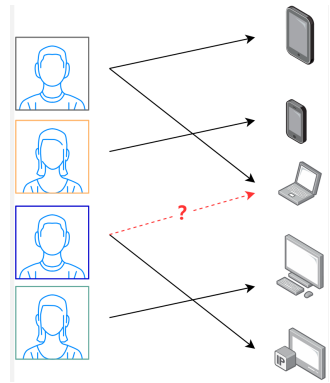
- A graph consists of a set of **nodes** $\mathcal{V}$ ($n = |\mathcal{V}|$) with corresponding node **features** denoted by $X \in \mathbb{R}^{n \times d}$ and set of **edges** $\mathcal{E}$[1]:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$$

- Graph tasks and applications:



(a) Community Detection



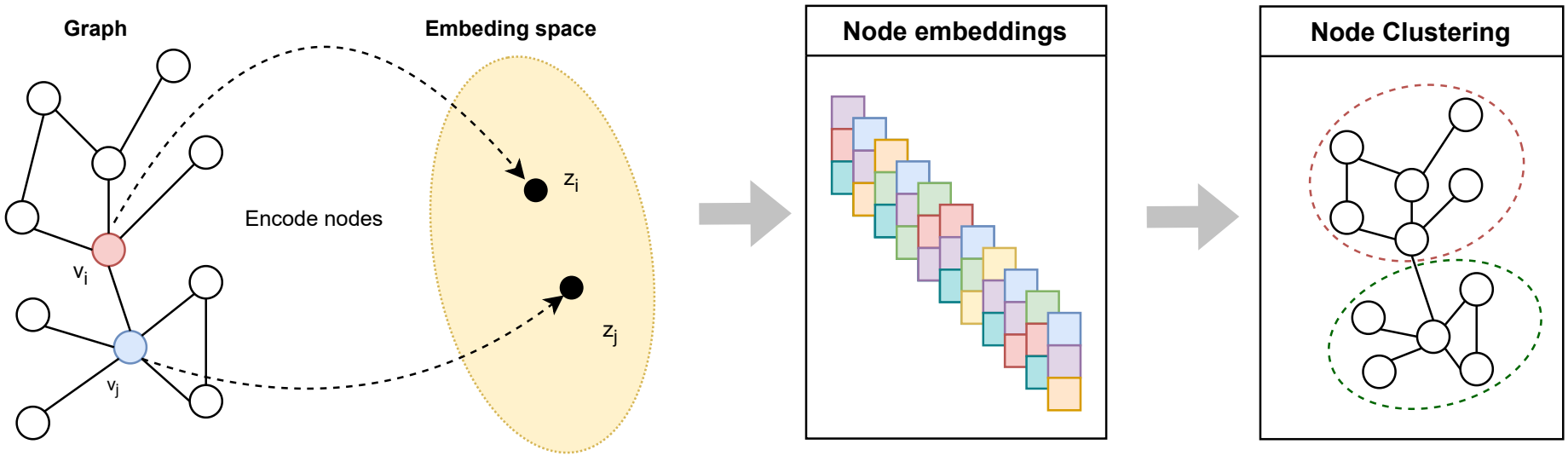
(b) Recommender System

[1] Here we focus on undirected graphs.

- Graph Representation Learning (GRL) seeks a function $\mathcal{F}$ to encode nodes of a graph into vectors:

$$\mathcal{F} : \mathcal{V} \rightarrow \mathbb{R}^m$$

- Nodes are mapped into a low-dimensional embedding space, where distances between nodes reflect their "relative positions" in the graph.



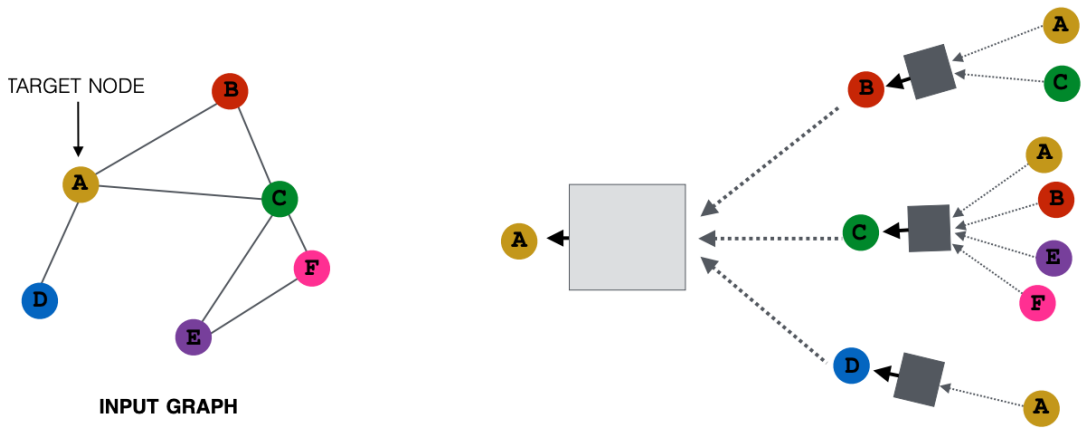
- Graph neural networks GNN provide a powerful paradigm for GRL with a recursive aggregation scheme[2][3]:
- The aggregation scheme can be summarized as:

$$\mathrm{GNN}\left(A, X, \left\{W^{(\ell)}\right\}\right) = H^{(t)},$$

where

  - $H^{(0)} = X$ and
  - $H^{(\ell)} = \sigma(agg(AH^{(\ell-1)}W^{(\ell)}))$ for all $\ell \in (0, t]$.

- Then we represent nodes using $\mathcal{F}(i) = H^{(t)}(i, \cdot)$.



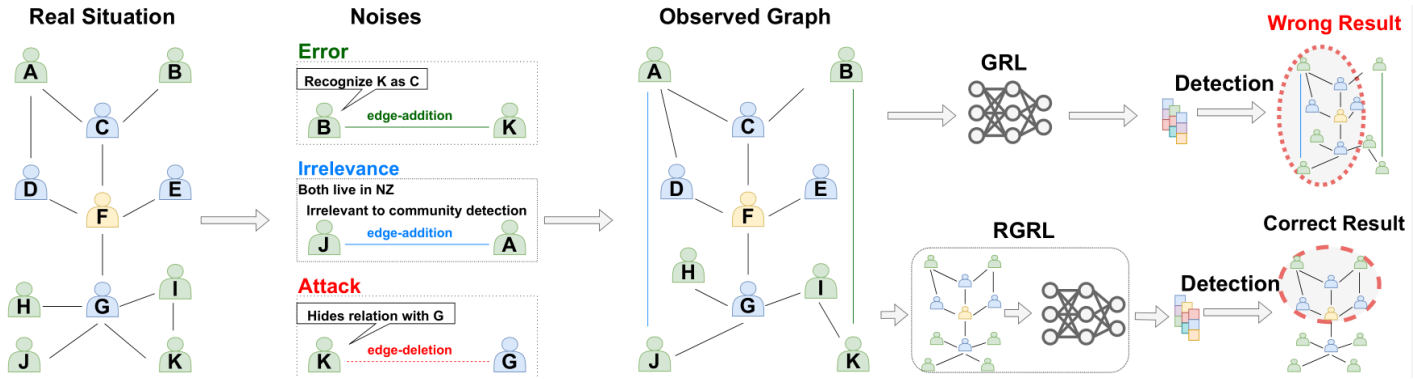
*https://snap-stanford.github.io/cs224w-notes

[2] William L. Hamilton, Zhitao Ying, and Jure Leskovec. NeurIPS (2017).

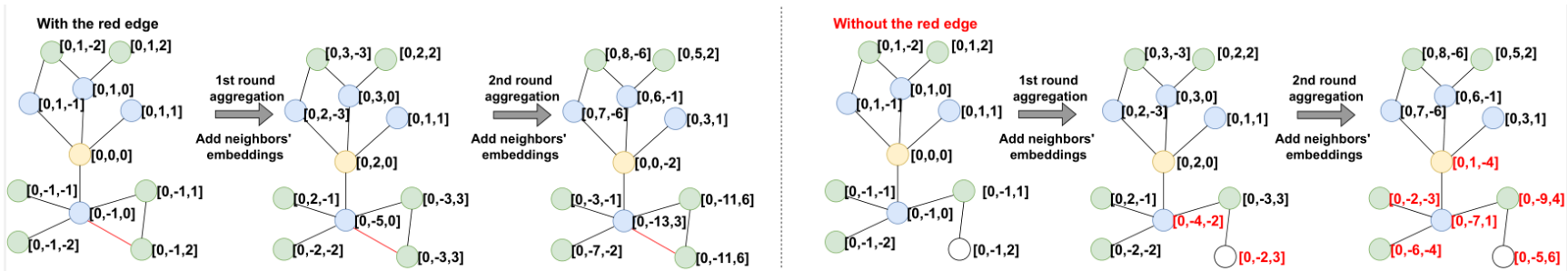
[3] Thomas N. Kipf and Max Welling. ICLR (2017).

Problem Formulation

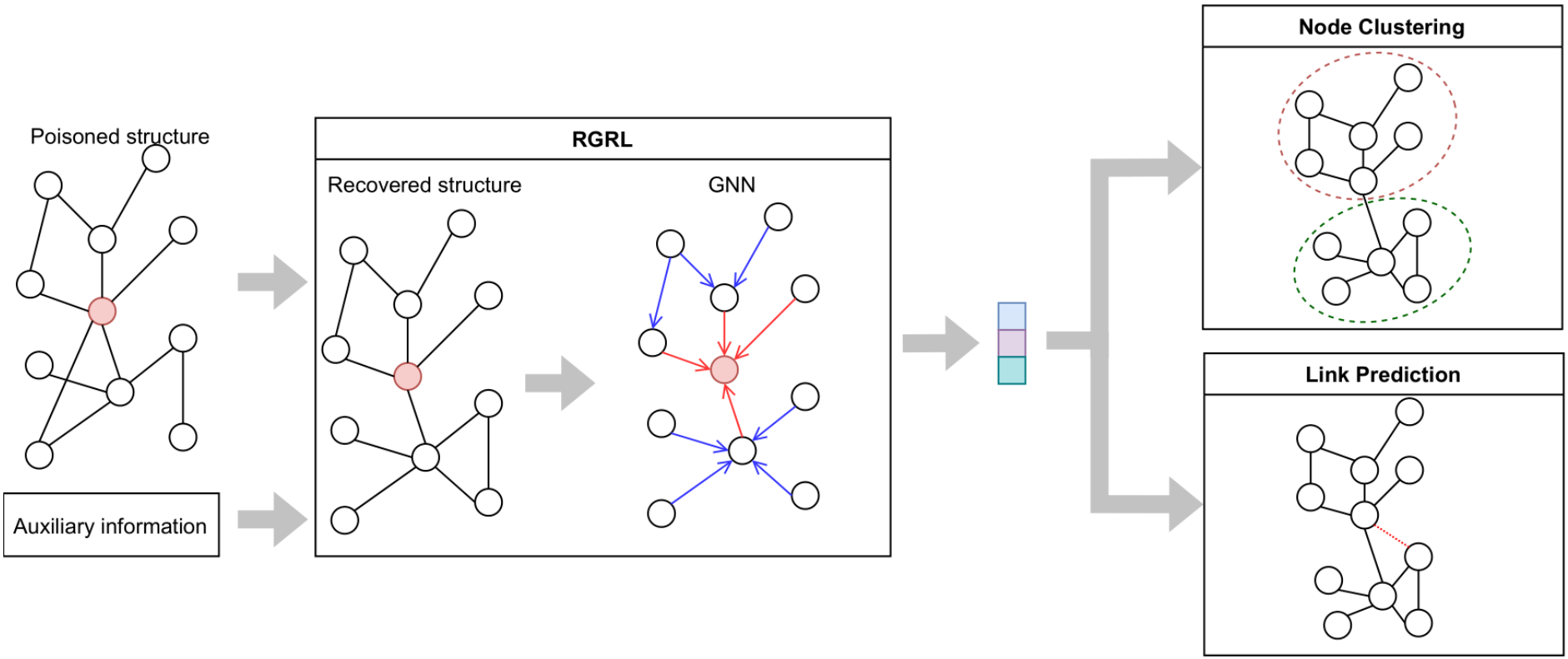
- Real-world graphs exhibit inherent uncertaintiesdue to data collection errors, attacks, etc.
- Such uncertainties bring perturbations such as (fake-)edge-additions and (real-)edge-deletion.



- These perturbations affect GNN's aggregation scheme and distort the resulting representations.



Robust graph representation learning (RGRL) aims to recover appropriate graph structure to get rid of the impact of perturbations when learning graph representations.

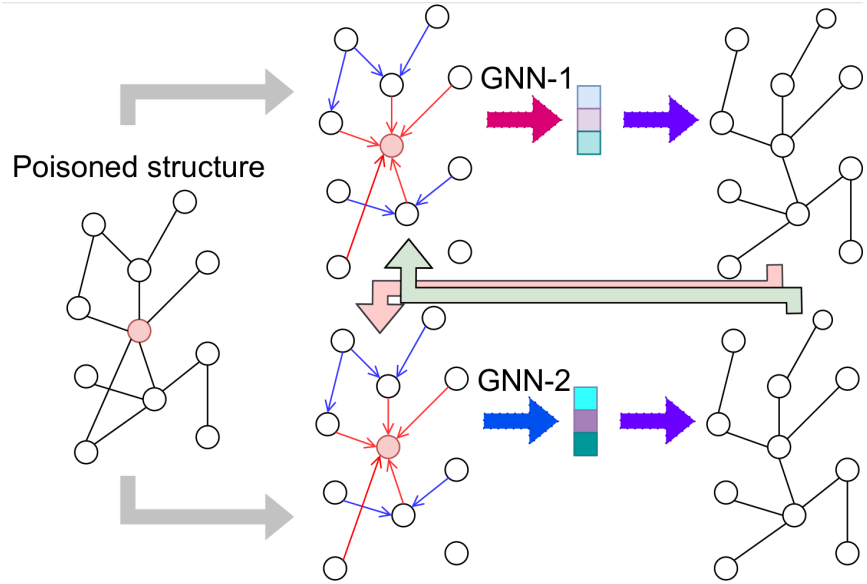


Supervised RGRL:

- 2019, GNN-Jaccard: remove fake edges[4].
- 2020, GNN-SVD: low-rank approximation[5].
- 2020, Pro-GNN: sparse, low-rank, and feature smoothing[6].

Unsupervised RGRL:

- 2020, Cross-Graph: peer-GNN structure[7].



[4] HuijunWu,ChenWang,YuriyTyshetskiy,AndrewDocherty,KaiLu,andLiming Zhu. IJCAI (2019).

[5] Negin Entezari, Saba Sayouri, Amirali Darvishzadeh, and Evangelos Papalexakis. WSDM (2020).

[6] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. KDD (2020).

[7] Chun Wang, Bo Han, Shirui Pan, Jing Jiang, Gang Niu, and Guodong Long. ICDM (2020).

In this paper we tackle the unsupervised RGRL problem.

Unsupervised Robust Graph Representation Learning

Alleviate the interference of random perturbations in the input graph while learning graph representations without label information.

Questions.

1. What are the properties of a graph G' that we could generate, which could help us mitigate the effects of undesirable perturbations?
2. How can we learn a model that could generate such a graph G'?

Method

Question 1: What are the properties of a graph G' that we could generate, which could help us mitigate the effects of undesirable perturbations?

We will define two properties:

1. Property 1: Innocuousness
2. Property 2: Local feature smoothness

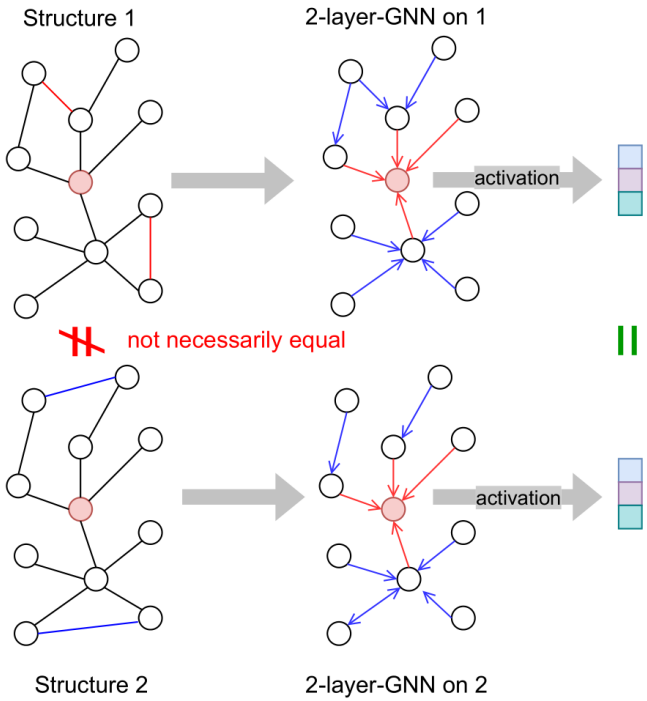
Property 1: Innocuousness

Definition

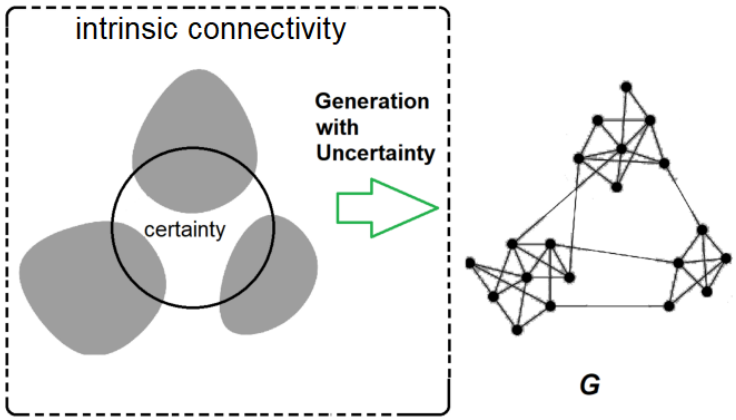
Given G_0 and G_1 with adjacency matrices A_0, A_2, resp. Suppose for each layer ℓ, for any input feature X, and any weight $W_0^{(\ell)}$, there exists weights $W_1^{(\ell)}$ such that

$$\sigma(\text{agg}(A_0 H^{(\ell-1)} W_0^{(\ell)})) = \sigma(\text{agg}(A_1 H^{(\ell-1)} W_1^{(\ell)})).$$

Then we call G_1 GNN-equivalent to G_0.



- Edges in a graph are formed with uncertainty, following some underlying intrinsic connectivity.
- We call those graphs that are GNN-equivalent to the intrinsic connectivity graph innocuous graphs.



Corollary

Let c be the rank of the adjacency matrix of the intrinsic connectivity graph. A graph G' is innocuous only if $\text{Rank}(A') \geq c$.

Property 1 (Innocuousness)

Suppose the number of connected components c of the intrinsic connectivity graph is given. Then we would like to find a graph G' with $\text{Rank}(A') \geq c$.

Property II: Local Feature-Smoothness

"*In a graph over which a GNN may extract semantically-useful node embeddings, adjacent nodes are likely to share similar features than non-adjacent nodes.*"

Suppose $f(\vec{x}, \vec{y})$ evaluates similarity between learnt node embeddings.

Property 2 (Local feature smoothness)

Take an arbitrary connected component C. Then for any three nodes v_a, v_b, v_c that satisfy $v_a \in C$, $v_b \in C$ and $v_c \notin C$, we have

$$f(X_a, X_b) \leq f(X_a, X_b)$$

Question 2: How can we learn a model that could generate such a graph G'?

1. For Property I: Network Partition Structural Information (NPSI)
2. For Property II: Davies-Bouldin index (DBI)

To Meet Property I

Definition [Li, Pan 2016; Liu, et al. 2019]

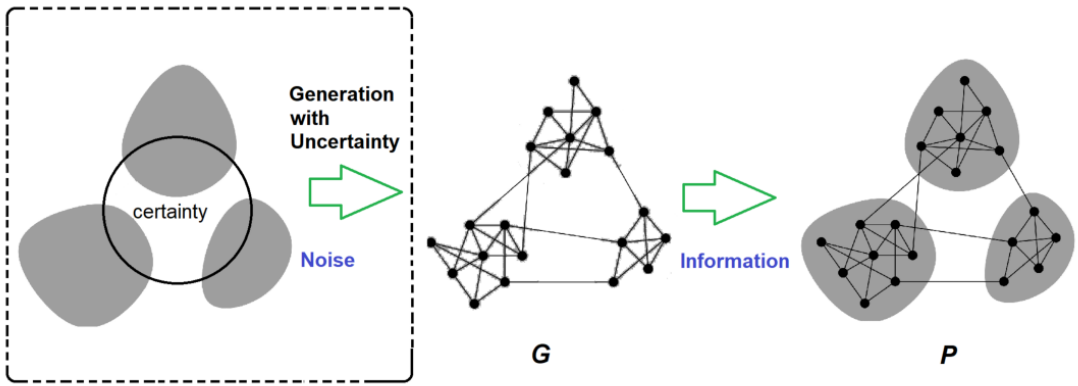
Given a partition P of a graph $\mathcal{G}'$ that consists of disjoint subgraph $C_1, \ldots, C_r$, the structural entropy of $\mathcal{G}'$ relative to P is

$$NPSI_{P(\mathcal{G}')} = \sum_{k<r} \left(\frac{vol_k - g_k}{2|\mathcal{E}'|} \log_2 \frac{vol_k}{2|\mathcal{E}'|} \right)$$

where

- vol_k is the number of edges which have some endpoint in C_k
- g_k is the number of edges which have only one endpoint in C_k

$NPSI_{P(\mathcal{G}')}$ captures the inherent randomness within the network partition P.



To utilize NPSI in GRL, we define a matrix form.

Matrix form of NPSI

- A': the adjacency matrix of $\mathcal{G}'$.
- $Y \in \{0,1\}^{n \times r}$: the indicator matrix with $Y_{(i,k)} = 1$ iff node i belongs to C_k.

Transform $NPSI_{P(\mathcal{G}')}$ into $NPSI(A', Y)$:

$$
\begin{aligned}
NPSI(A', Y) &= \sum_{k<r} \left(\frac{vol_k - g_k}{2|\mathcal{E}|} \log_2 \frac{vol_k}{2|\mathcal{E}|} \right) \\
&= \sum_{k<r} \left(\frac{(Y^T A' Y)_{kk}}{2sum(A')} \times \log_2 \left(\frac{(\{1\}^{r \times n} A' Y)_{kk}}{2sum(A')} \right) \right) \\
&= trace \left(\frac{Y^T A' Y}{2sum(A')} \otimes \log_2 \left(\frac{\{1\}^{r \times n} AY}{2sum(A')} \right) \right) = NPSI_{P(\mathcal{G}')}
\end{aligned}
$$

Theorem

For a given indicator matrix $Y \in \{0,1\}^{n \times r}$. Suppose $A = \arg\min_A NPSI(A, Y)$ such that $A_{ij} \geq 0$ for any i, j and $A = A^{\mathsf{T}}$. Then $\text{Rank}(A) \geq r$.

To meet Property I

We search for an innocuous graph by minimizing $NPSI(A, Y)$ using a (learned) Y with $r = c$.

To Meet Property II

Davies-Bouldin index (DBI)

DBI is used to analyze the similarity of node features:

$$DBI(X, Y) = \frac{1}{r} \sum_{k<r} DI_k$$

$$\text{where: } DI_k = max_{m \neq k}(R_{km}) \text{ , } R_{km} = \frac{S_k + S_m}{M_{km}}$$

$$S_k = (\frac{1}{|C_k|} \sum_{Y_{ik}=1} (|X_i - \overline{X}_k|^2))^{\frac{1}{2}},$$

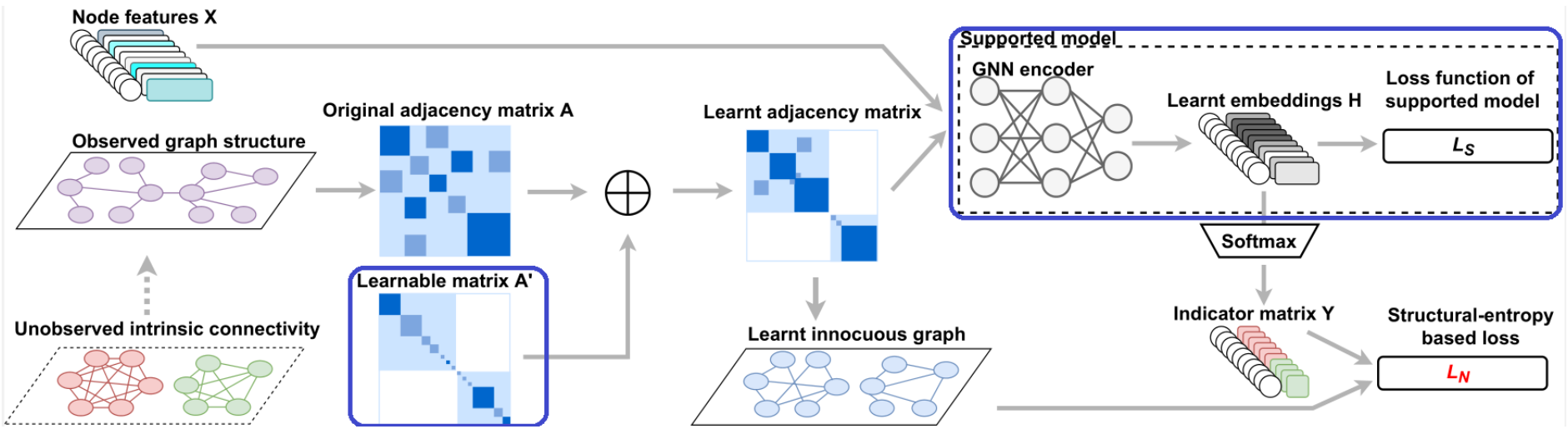
$$M_{km} = (|\overline{X}_k - \overline{X}_m|^2)^{\frac{1}{2}} \text{ , } \overline{X}_k = \frac{\sum_{Y_{ik}=1} X_i}{|C_k|}.$$

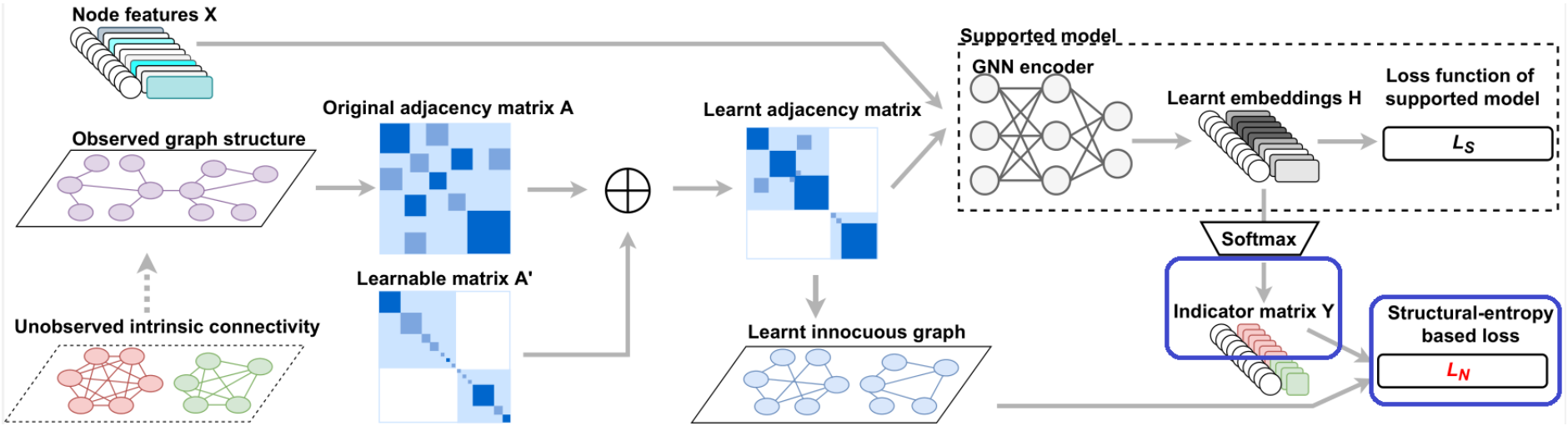
To Meet Property II

We would minimize $DBI(X, Y)$.

Unsupervised Structural Entropy-based Robust GNN

- The learnable A' is added to learn innocuous graph.
- The supported model can be any GNN model.





- To learn Y: Fix the current A', minimize NPSI:

$$Y = \arg\min_{Y} \left(NPSI(A', Y)\right)$$

$$s.t.\ Y \in \{0,1\}^{n\times r},\ (Y^T Y)_{km} \begin{cases} > 0 & \text{if } k = m, \\ = 0 & \text{otherwise.} \end{cases}$$

- To learn A': Fix the current Y, minimize the loss $\mathcal{L}_N$:

$$\mathcal{L}_N = NPSI(A', Y) + \beta DBI(X, Y)$$

Experiments

- Datasets: Cora, Citeseer, Wiki
- Perturbation: Random, Meta-attack (Zügner, Günnemann 2019)
- Supported model: GAE (Kipf, Welling 2016)
- Baselines: 10 models
- Tasks:
  - Node clustering
  - Link prediction

Experiments: Node Clustering

Table: Node clustering performance (NMI ± Std) under random-noises

Dataset	Ptb Rate (%)	GAE	AGE	DGI	GIC	GCA	GAE_CG	ARGA_CG	USER
cora	10	41.28 ± 2.11	42.23 ± 0.55	53.45 ± 1.0	50.24 ± 0.69	35.1 ± 2.33	42.47 ± 1.87	43.11 ± 2.6	54.38 ± 2.23
	20	33.0 ± 2.91	35.1 ± 0.61	50.47 ± 0.78	48.54 ± 0.69	32.18 ± 3.05	38.31 ± 3.07	37.96 ± 1.89	52.17 ± 1.94
	30	29.07 ± 4.38	36.05 ± 1.04	48.22 ± 0.48	44.75 ± 0.57	32.93 ± 2.23	35.19 ± 1.99	36.59 ± 1.47	52.39 ± 1.0
	40	27.08 ± 2.53	32.68 ± 0.63	44.02 ± 1.26	40.97 ± 0.96	33.4 ± 1.73	33.47 ± 2.38	34.46 ± 2.23	46.7 ± 2.7
	50	25.03 ± 3.05	36.79 ± 2.81	43.22 ± 0.69	40.73 ± 1.0	31.81 ± 1.65	31.93 ± 1.86	31.53 ± 2.72	49.33 ± 1.95
citeseer	10	18.3 ± 2.69	29.47 ± 1.85	41.31 ± 0.72	41.29 ± 0.76	10.75 ± 2.2	20.41 ± 1.9	18.26 ± 2.54	37.04 ± 1.46
	20	16.16 ± 1.62	21.65 ± 0.74	36.66 ± 0.71	36.72 ± 0.91	7.0 ± 0.65	16.84 ± 1.77	16.93 ± 1.19	34.42 ± 3.35
	30	12.86 ± 1.75	18.06 ± 1.22	33.39 ± 0.78	33.58 ± 0.95	6.16 ± 1.3	15.17 ± 1.63	14.41 ± 1.37	34.5 ± 3.06
	40	9.81 ± 2.52	15.57 ± 1.11	32.26 ± 0.73	31.91 ± 0.89	4.22 ± 0.88	12.71 ± 1.4	12.03 ± 2.31	34.58 ± 2.49
	50	10.41 ± 1.73	14.13 ± 0.91	29.58 ± 0.94	30.96 ± 0.77	2.98 ± 0.35	12.59 ± 2.83	13.66 ± 1.71	34.5 ± 2.1
wiki	10	22.57 ± 8.43	48.85 ± 1.01	40.13 ± 0.71	24.65 ± 1.43	30.77 ± 1.62	19.69 ± 8.96	19.63 ± 9.06	48.97 ± 1.16
	20	13.66 ± 7.96	46.92 ± 0.57	36.15 ± 1.36	22.9 ± 1.66	30.74 ± 1.09	11.62 ± 5.32	16.54 ± 7.83	48.71 ± 1.63
	30	14.7 ± 4.2	47.43 ± 0.65	34.84 ± 0.68	38.29 ± 0.68	31.42 ± 1.64	16.5 ± 6.29	10.66 ± 5.27	48.55 ± 1.44
	40	8.26 ± 7.51	46.7 ± 0.59	31.28 ± 1.25	36.33 ± 0.54	32.38 ± 1.39	9.99 ± 10.57	11.41 ± 6.2	48.54 ± 2.02
	50	9.52 ± 6.89	46.83 ± 0.58	29.29 ± 1.15	33.26 ± 0.89	34.24 ± 1.75	9.26 ± 4.87	6.6 ± 3.32	48.68 ± 1.78

Table: Node clustering performance (NMI ± Std) under meta-attack

Dataset	Ptb (%)	GAE	AGE	DGI	GIC	GCA	GAE_CG	ARGA_CG	USER
cora	5	43.37 ± 3.34	48.6 ± 1.73	50.33 ± 2.3	46.89 ± 2.05	38.12 ± 3.46	43.64 ± 3.44	43.0 ± 3.15	50.64 ± 2.77
	10	34.1 ± 3.44	39.35 ± 3.14	37.73 ± 3.63	36.58 ± 3.11	34.07 ± 2.77	35.47 ± 2.79	35.94 ± 3.51	41.71 ± 3.32
	15	19.96 ± 4.11	25.39 ± 3.88	23.13 ± 3.39	23.19 ± 3.29	21.54 ± 4.61	22.59 ± 3.69	22.92 ± 3.38	29.27 ± 3.68
	20	7.26 ± 2.86	9.65 ± 3.35	10.17 ± 2.67	10.96 ± 3.11	9.97 ± 2.01	10.34 ± 3.2	10.31 ± 2.96	18.82 ± 2.9
citeseer	5	22.5 ± 3.43	34.06 ± 1.99	40.22 ± 1.89	39.91 ± 1.95	20.78 ± 5.93	22.67 ± 2.54	21.69 ± 3.07	35.72 ± 2.03
	10	22.25 ± 2.6	25.13 ± 2.7	29.71 ± 3.05	29.45 ± 3.0	18.92 ± 1.91	22.6 ± 1.94	22.06 ± 1.87	31.86 ± 2.84
	15	13.73 ± 2.81	15.71 ± 2.01	17.68 ± 2.77	17.81 ± 2.68	13.61 ± 1.99	15.6 ± 2.35	15.61 ± 2.15	27.77 ± 3.31
	20	5.64 ± 2.01	9.11 ± 0.85	9.11 ± 1.78	9.08 ± 2.17	7.08 ± 2.03	7.68 ± 2.06	7.61 ± 2.12	26.42 ± 2.67
wiki	5	19.59 ± 7.49	41.76 ± 1.31	32.94 ± 2.61	35.03 ± 3.18	27.24 ± 1.4	18.24 ± 7.97	16.27 ± 5.05	48.44 ± 1.71
	10	13.09 ± 6.62	38.72 ± 0.26	22.59 ± 3.1	23.64 ± 2.65	25.86 ± 1.81	13.34 ± 6.63	10.98 ± 4.41	47.71 ± 1.7
	15	4.59 ± 5.02	40.9 ± 0.89	12.27 ± 2.43	15.19 ± 0.93	20.14 ± 5.08	4.62 ± 5.21	7.04 ± 3.79	47.54 ± 1.53
	20	2.22 ± 3.4	42.71 ± 0.98	8.85 ± 0.77	9.24 ± 2.33	15.39 ± 2.7	3.1 ± 3.9	2.49 ± 0.21	47.48 ± 1.54

Experiments: Link Prediction Performance

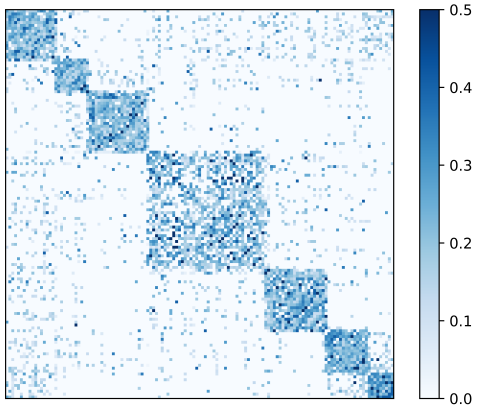
Table: Link prediction (AUC ± Std) under random-noises

Dataset	Ptb Rate (%)	GAE	ARGA	GAE_CG	ARGA_CG	USER
citeseer	0	94.09 ± 0.26	94.87 ± 0.48	94.08 ± 0.63	94.0 ± 0.88	95.38 ± 0.12
	10	94.09 ± 0.78	94.25 ± 0.21	93.96 ± 0.41	94.04 ± 1.03	95.59 ± 0.16
	20	94.12 ± 0.09	93.69 ± 1.16	94.01 ± 0.44	94.05 ± 0.34	94.99 ± 0.65
	30	91.72 ± 0.2	92.31 ± 0.82	93.15 ± 0.1	93.27 ± 0.17	95.15 ± 0.17
	40	90.29 ± 2.99	90.81 ± 0.53	93.07 ± 0.79	92.89 ± 0.46	94.41 ± 0.21
	50	90.05 ± 1.95	90.91 ± 0.51	91.85 ± 0.87	91.6 ± 0.12	94.54 ± 0.24
wiki	0	86.75 ± 1.05	82.14 ± 15.09	83.25 ± 7.29	79.81 ± 6.89	88.72 ± 0.14
	10	80.12 ± 16.61	83.66 ± 6.24	68.04 ± 12.66	77.24 ± 4.86	88.07 ± 0.32
	20	79.5 ± 4.86	80.86 ± 10.5	70.62 ± 6.33	74.57 ± 3.44	87.82 ± 0.27
	30	73.02 ± 10.44	80.06 ± 3.96	66.27 ± 9.57	68.97 ± 6.42	87.41 ± 0.51
	40	78.37 ± 4.91	79.44 ± 4.46	61.58 ± 6.22	64.06 ± 9.79	87.45 ± 0.22
	50	67.78 ± 4.55	76.79 ± 2.16	64.48 ± 9.48	71.51 ± 8.56	86.87 ± 0.4

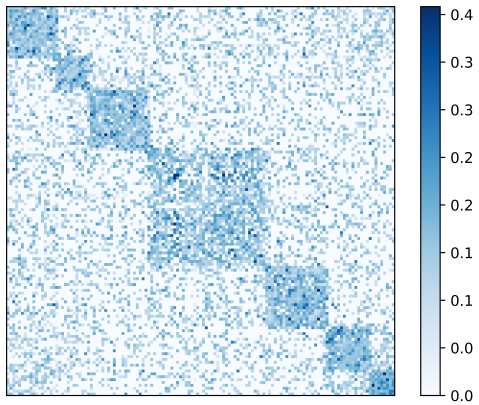
Table: Link prediction (AP ± Std) under random-noises

Dataset	Ptb Rate (%)	GAE	ARGA	GAE_CG	ARGA_CG	USER
citeseer	0	94.22 ± 0.32	94.89 ± 0.5	94.5 ± 0.29	94.14 ± 0.74	95.84 ± 0.08
	10	94.08 ± 0.94	94.47 ± 0.44	94.21 ± 0.41	94.38 ± 1.19	95.98 ± 0.08
	20	94.57 ± 0.01	94.05 ± 1.14	94.5 ± 0.24	94.42 ± 0.51	95.46 ± 0.67
	30	92.13 ± 0.21	92.77 ± 0.93	93.83 ± 0.32	93.94 ± 0.17	95.66 ± 0.17
	40	90.95 ± 2.81	91.38 ± 0.33	93.62 ± 0.61	93.58 ± 0.25	94.92 ± 0.16
	50	90.81 ± 2.12	91.66 ± 0.63	92.73 ± 0.71	92.44 ± 0.28	95.05 ± 0.19
wiki	0	88.18 ± 1.49	83.38 ± 17.68	85.65 ± 7.11	83.01 ± 6.33	89.9 ± 0.1
	10	81.95 ± 17.34	86.02 ± 4.96	69.8 ± 14.58	80.78 ± 4.35	89.48 ± 0.22
	20	82.62 ± 3.11	82.44 ± 12.9	73.09 ± 6.73	77.22 ± 3.33	89.07 ± 0.27
	30	76.89 ± 12.98	82.98 ± 2.6	69.06 ± 11.63	72.13 ± 6.08	88.9 ± 0.27
	40	81.53 ± 3.1	82.22 ± 2.94	63.15 ± 7.47	65.91 ± 12.33	88.61 ± 0.08
	50	70.36 ± 2.81	80.29 ± 1.83	67.09 ± 10.51	73.6 ± 11.85	88.28 ± 0.38

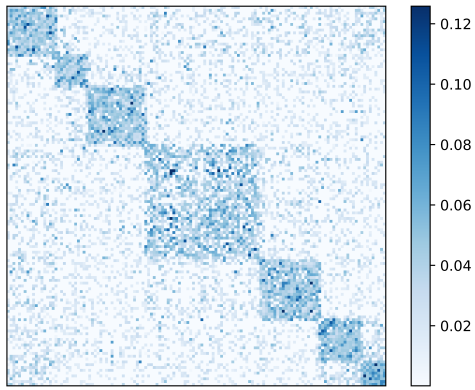
Case Study



(c) Original



(d) Poisoned



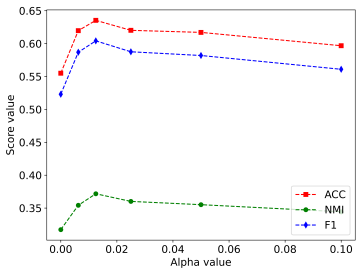
(e) USER

Figure: The graph structure heat maps of Cora

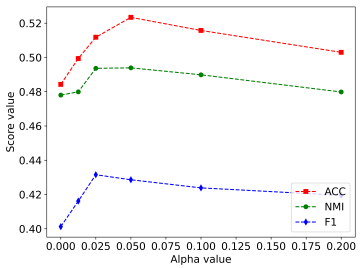
Ablation Study and Parameter Analysis

Table: NMI of USER's variants with 10% random-noise

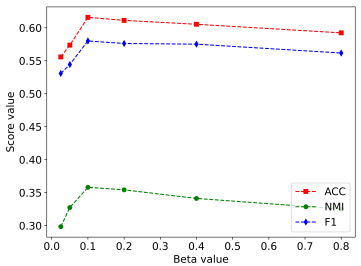
Dataset	USER	w.o. NPSI	w.o. DBI	Fix A'
cora	54.38	14.82	52.54	40.11
citeseer	37.04	28.95	12.82	30.94
wiki	48.97	48.44	37.28	39.77



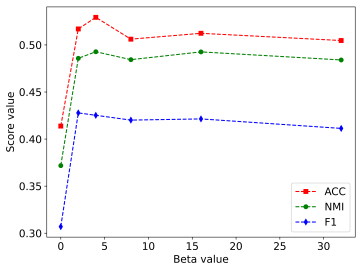
(a) α on citeseer



(b) α on wiki



(c) β on citeseer



(d) β on wiki

Summary of Contribution

1. Propose USER model for unsupervised RGRL.
2. Properties of graphs to be generated that mitigates the effect of perturbation.
   - Innocuousness
   - Local feature smoothness
3. Optimization criteria to meet the desired properties:
   - Structural entropy
   - DBI

Code is available: `https://github.com/wangyifeibeijing/USER`

Thank you!