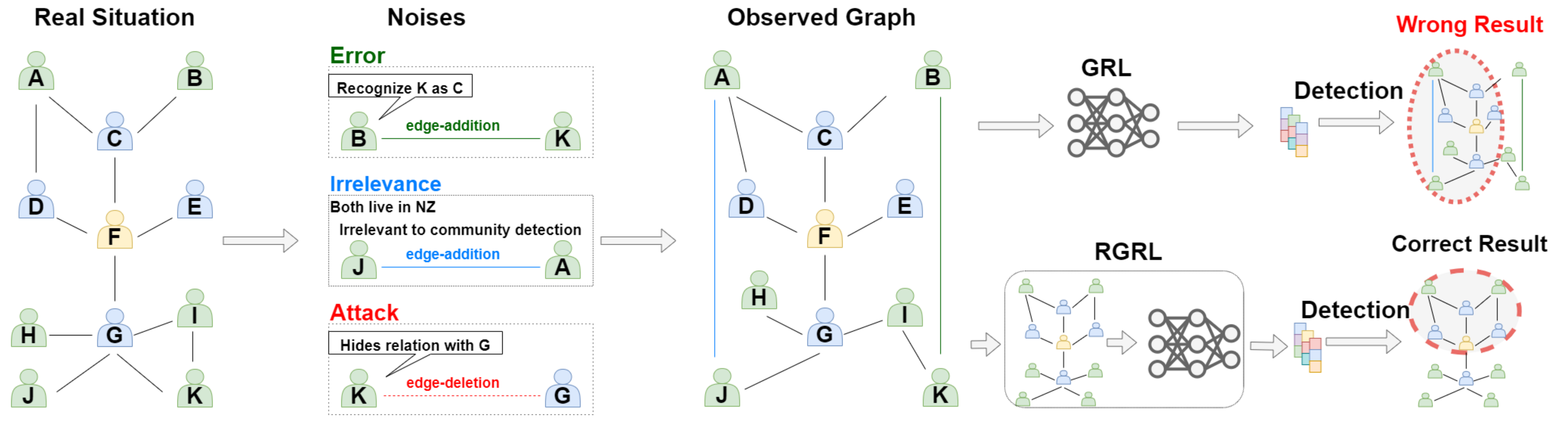
USER: Unsupervised Structural Entropy-based Robust Graph Neural Network

Yifei Wang, Yupan Wang, Zeyu Zhang, Song Yang, Kaiqi Zhao, Jiamou Liu
The University of Auckland, Auckland, New Zealand
wany107@aucklanduni.ac.nz, jiamou.liu@auckland.ac.nz

Background

- Unsupervised graph representation learning models are vulnerable to inherent randomness in the input graph.
- ***Unsupervised robust graph representation learning*** aims to alleviate the interference of randomness in the input graph while learning graph representation without label information.



Challenges

- How to find a graph that mitigates the interference of randomness in the input data without labels?
- What's objective function that guides a model to reveal such graphs?

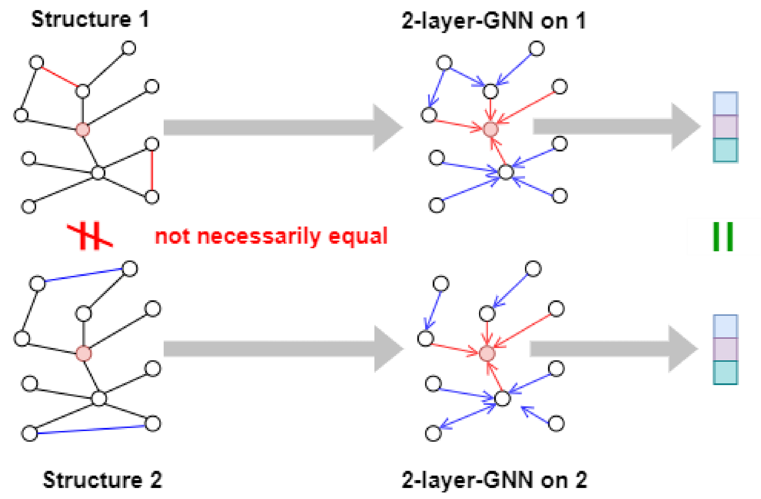
Main Idea

Idea: Learn **innocuous graphs** that are **GNN-equivalent** to an unobserved **intrinsic connectivity graph** and **locally feature smooth**

- Edges in a graph are formed randomly, following certain underlying **intrinsic connectivity**.
- Two graphs with which GNN would learn the same embeddings are not necessarily equal. We say that they are **GNN-equivalent**.
- Although the intrinsic connectivity is unobserved, certain graphs are GNN-equivalent to it, i.e., the **innocuous graphs**.

Theorem 1. (Necessary condition of GNN-equivalence) G_1 is GNN-equivalent to G_0 only if $Rank(A_1) \geq Rank(A_0)$

Corollary 1. let c be the rank of the intrinsic connectivity graph, G' is an innocuous graph only if $Rank(A') \geq c$.



- Let X be the feature matrix
- $f(\vec{x}, \vec{y})$: a function that evaluates distance between vectors
- C: a connected component

- **Local feature smoothness:** For nodes $v_a \in C$, $v_b \in C$ and $v_c \notin C$, $f(X_a, X_b) \leq f(X_a, X_c)$.

Structural Entropy-based Objective Function

To search for innocuous graphs, we invoke ***network partition structural information (NPSI)*** [1,2]:

$$NPSI_{\mathrm{GP}(G)} := \sum_{k<r} \frac{vol_k - g_k}{2|\varepsilon|} \log_2 \frac{vol_k}{2|\varepsilon|},$$

where vol_k is volumn of C_k and g_k denotes the number of edges with exactly one end in C_k.
To utilize it in GNN models, we introduce a matrix form of NPSI:

$$NPSI(\mathbf{A}, \mathbf{Y}) := \mathrm{trace}\left(\frac{Y^T A Y}{2sum(A)} \otimes \log_2\left(\frac{\{1\}^{r\times n} A Y}{2sum(A)}\right)\right),$$

where A is the adjacency matrix of graph G while $Y \in \{0,1\}^{n\times r}$ is an indicator matrix that satisfies $Y_{ik} = 1$ if $i \in C_k$, otherwise $Y_{ik} = 0$.

Theorem 2. (minimize $NPSI(A, Y)$ with learnable A):
Suppose $A = \operatorname*{argmin}_A NPSI(A, Y)$, s.t., $A_{ij} \geq 0$ and $A = A^T$. Then $Rank(A) \geq r$.

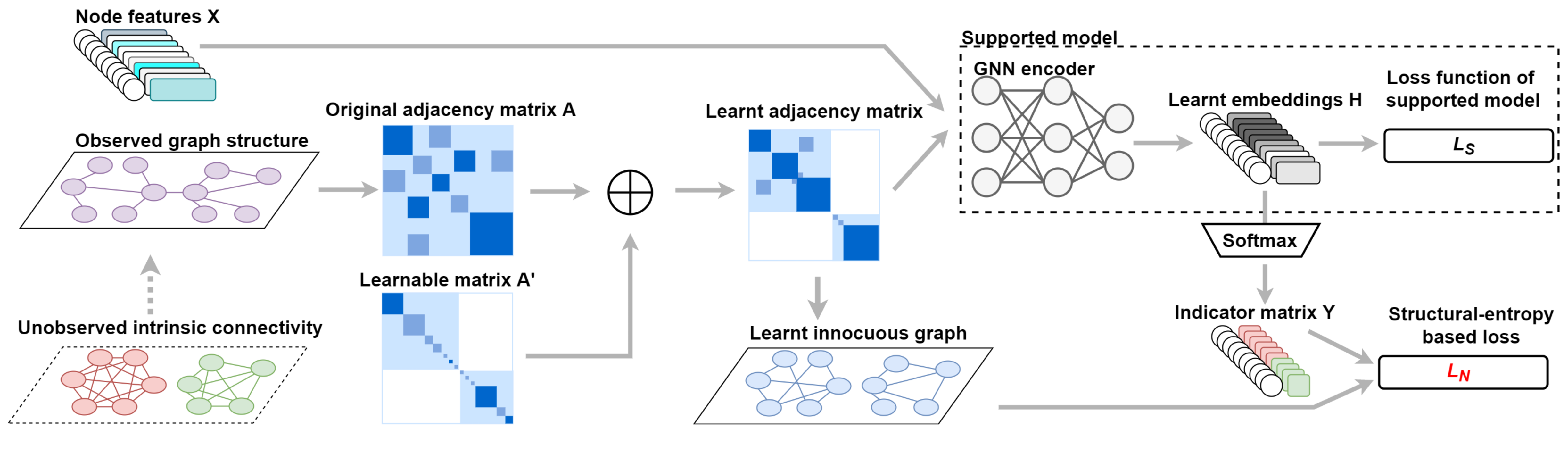
By **Theorem 1** and **Theorem 2**, one may search for a graph that satisfies the necessary condition of being innocuous graph **(Corollary 1)** by minimizing $NPSI(A, Y)$ with $r = c$.

With $\{C_1, C_2 \cdots C_r\}$ indicated by Y, a graph satisfies **local feature smoothness** can be learned by minimizing **Davies-Bouldin index $DBI(X, Y)$**, where $Y = \operatorname*{argmin}_{A,Y} NPSI(A, Y)$ [3].

The process of finding an innocuous graph into an optimization requires minimizing the **loss L_N** :

$$L_N := NPSI(\mathbf{A}, \mathbf{Y}) + \beta DBI(X, Y)$$

Unsupervised Structural Entropy-based Robust GNN



USER framework facilitates GNN models to learn embeddings and the innocuous adjacency matrix simultaneously. :

- ➢ ***A' : a learnable matrix***
- ➢ ***GNN encoder***: a ***supported model***, can be any unsupervised GNN model
- ➢ **L_S**: the loss function of the supported model

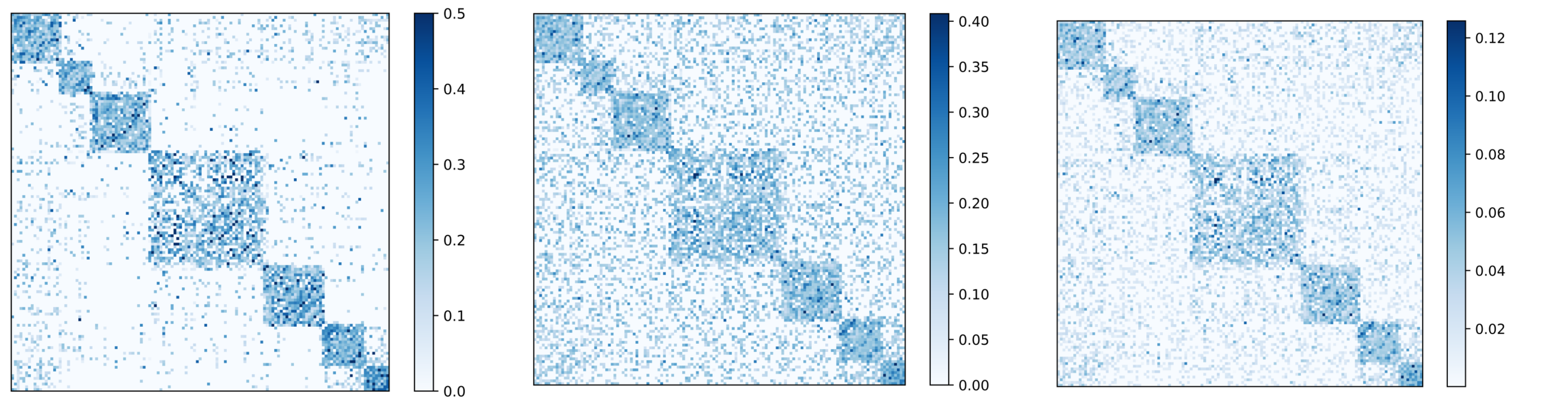
Experiments

Dataset: 3 real-world networks. **Noises:** {Random noise, meta-attack}.
Tasks: {Node clustering, Link prediction}.

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

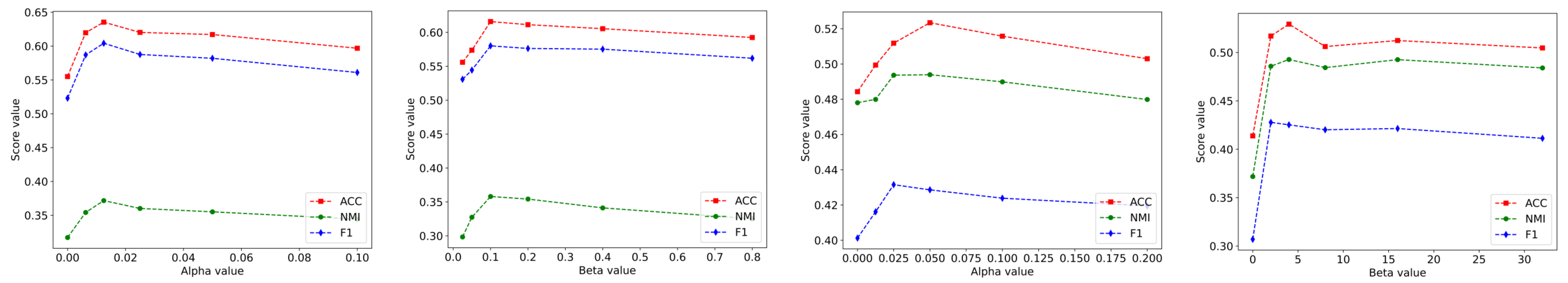
Dataset	Ptb Rate (%)	deepwalk	GAE	VGAE	ARGA	AGE	DGI	GIC	GCA	GAE_CG	ARGA_CG	USER
cora	5	41.73 ± 2.16	43.37 ± 3.34	43.06 ± 2.66	43.33 ± 3.28	48.6 ± 1.73	50.33 ± 2.3	46.89 ± 2.05	38.12 ± 3.46	43.64 ± 3.44	43.0 ± 3.15	$\mathbf{50.64 \pm 2.77}$
	10	37.68 ± 2.86	34.1 ± 3.44	33.6 ± 3.66	34.5 ± 3.71	39.35 ± 3.14	37.73 ± 3.63	36.58 ± 3.11	34.07 ± 2.77	35.47 ± 2.79	35.94 ± 3.51	$\mathbf{41.71 \pm 3.32}$
	15	21.99 ± 4.38	19.96 ± 4.11	19.56 ± 4.18	20.04 ± 3.82	25.39 ± 3.88	23.13 ± 3.39	23.19 ± 3.29	21.54 ± 4.61	22.59 ± 3.69	22.92 ± 3.38	$\mathbf{29.27 \pm 3.68}$
	20	7.31 ± 2.85	7.26 ± 2.86	7.22 ± 2.91	7.88 ± 2.79	9.65 ± 3.35	10.17 ± 2.67	10.96 ± 3.11	9.97 ± 2.01	10.34 ± 3.2	10.31 ± 2.96	$\mathbf{18.82 \pm 2.9}$
citeseer	5	16.97 ± 2.14	22.5 ± 3.43	22.73 ± 2.68	20.85 ± 2.69	34.06 ± 1.99	$\mathbf{40.22 \pm 1.89}$	39.91 ± 1.95	20.78 ± 5.93	22.67 ± 2.54	21.69 ± 3.07	35.72 ± 2.03
	10	23.52 ± 1.69	22.25 ± 2.6	22.59 ± 2.62	22.02 ± 2.22	25.13 ± 2.7	29.71 ± 3.05	29.45 ± 3.0	18.92 ± 1.91	22.6 ± 1.94	22.06 ± 1.87	$\mathbf{31.86 \pm 2.84}$
	15	17.33 ± 2.73	13.73 ± 2.81	13.6 ± 2.95	13.94 ± 2.65	15.71 ± 2.01	17.68 ± 2.77	17.81 ± 2.68	13.61 ± 1.99	15.6 ± 2.35	15.61 ± 2.15	$\mathbf{27.77 \pm 3.31}$
	20	8.3 ± 2.45	5.64 ± 2.01	5.71 ± 1.82	5.63 ± 1.74	9.11 ± 0.85	9.11 ± 1.78	9.08 ± 2.17	7.08 ± 2.03	7.68 ± 2.06	7.61 ± 2.12	$\mathbf{26.42 \pm 2.67}$
wiki	5	34.06 ± 1.74	19.59 ± 7.49	19.22 ± 7.6	20.8 ± 5.9	41.76 ± 1.31	32.94 ± 2.61	35.03 ± 3.18	27.24 ± 1.4	18.24 ± 7.97	16.27 ± 5.05	$\mathbf{48.44 \pm 1.71}$
	10	22.96 ± 2.74	13.09 ± 6.62	11.14 ± 5.52	12.48 ± 4.46	38.72 ± 0.26	22.59 ± 3.1	23.64 ± 2.65	25.86 ± 1.81	13.34 ± 6.63	10.98 ± 4.41	$\mathbf{47.71 \pm 1.7}$
	15	14.35 ± 1.8	4.59 ± 5.02	4.99 ± 4.31	6.82 ± 3.15	40.9 ± 0.89	12.27 ± 2.43	15.19 ± 0.93	20.14 ± 5.08	4.62 ± 5.21	7.04 ± 3.79	$\mathbf{47.54 \pm 1.53}$
	20	9.3 ± 1.2	2.22 ± 3.4	1.52 ± 3.09	3.61 ± 1.85	42.71 ± 0.98	8.85 ± 0.77	9.24 ± 2.33	15.39 ± 2.7	3.1 ± 3.9	2.49 ± 0.21	$\mathbf{47.48 \pm 1.54}$

Case study of learnt innocuous graph



(a) Original (b) Poisoned (c) USER

Parameter analysis



(a) α on Citeseer (b) β on Citeseer (c) α on Wiki (b) β on Wiki

Conclusions

- ➢ There are multiple innocuous graphs with which GNN can learn the appropriate embeddings.
- ➢ NPSI-based objective function is an ideal tool to capture innocuous graph.
- ➢ Unsupervised RGRL framework may effectively alleviate the interference of graph randomness.
- ➢ Code available: https://github.com/wangyifeibeijing/USER

[1] Li, A., & Pan, Y. (2016) Structural information and dynamical complexity of networks.
[2] Liu, Y., Liu, J., Zhang, Z., Zhu, L., & Li, A. (2019). REM: From structural entropy to community structure deception
[3] Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure.