SE-GSL: A General and Effective Graph Structure Learning Framework through Structural Entropy Optimization

Dongcheng Zou
Beihang University
Beijing, China
zoudongcheng@buaa.edu.cn

Hao Peng*
Beihang University
Beijing, China
penghao@buaa.edu.cn

Xiang Huang
Beihang University
Beijing, China
huang.xiang@buaa.edu.cn

Renyu Yang
Beihang University
Beijing, China
renyu.yang@buaa.edu.cn

Jianxin Li
Beihang University
Beijing, China
lijx@buaa.edu.cn

Jia Wu
Macquarie University
Sydney, Australia
jia.wu@mq.edu.au

Chunyang Liu
Didi Chuxing
Beijing, China
liuchunyang@didiglobal.com

Philip S. Yu
University of Illinois Chicago
Chicago, USA
psyu@uic.edu

WWW 2023

code: https://github.com/RingBDStack/SE-GSL

Reported by Zicong Dou

Introduction

i) robustness to system noises and heterophily graphs
ii) model interpretability.

This paper proposes a general GSL framework, SE-GSL, through structural entropy and the graph hierarchy abstracted in the encoding tree.

Graph and Community

$G = \{V, E, X\}$ denote a graph

$X \in \mathbb{R}^{n \times d}$ refers to the vertex attribute set.

$A \in \mathbb{R}^{n \times n}$ $\quad d(v_i) = \sum_j A_{ij}$

$D = \mathrm{diag}(d(v_1), d(v_2), \ldots, d(v_n))$

$\mathcal{P} = \{P_1, P_2, \ldots, P_L\}$ is a partition of V

Graph Structure Learning (GSL)

$$\mathcal{L}_{gsl} = \mathcal{L}_{task}(Z, Y) + \alpha \mathcal{L}_{reg}(Z, G^*, G)$$

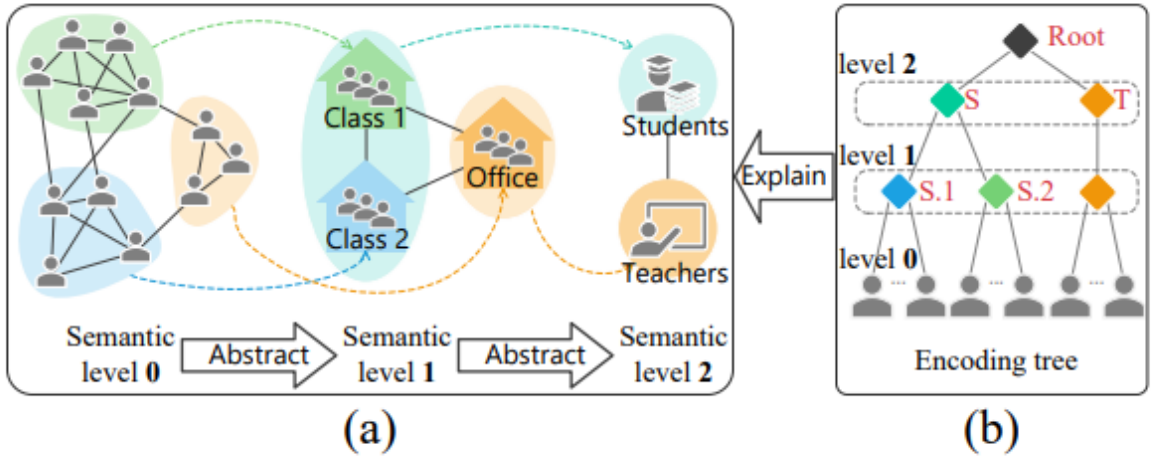
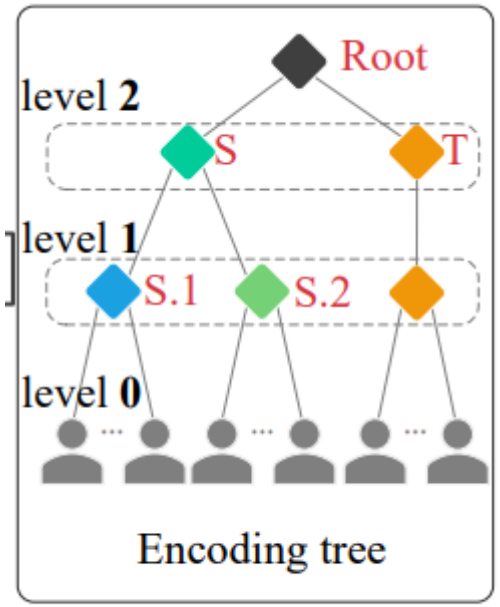


Figure 1: An illustrative example of the hierarchical community (semantics) in a simple social network. (1) Vertices and edges represent the people and their interconnectivity (e.g., common locations, interests, occupations). There are different abstraction levels, and each community can be divided into sub-communities in a finer-grained manner (e.g., students are placed in different classrooms while teachers are allocated different offices). The lowest abstraction will come down to the individuals with own attributes, and the highest abstraction is the social network system. (b) An encoding tree is a natural form to represent and interpret such a multi-level hierarchy.

Introduction



(b)

Encoding Tree

the encoding tree $\mathcal{T}$ of graph $G = (V, E)$

The root node λ in $\mathcal{T}$ has a label $T_\lambda = V$

Each non-root node α has a label $T_\alpha \subset V$.

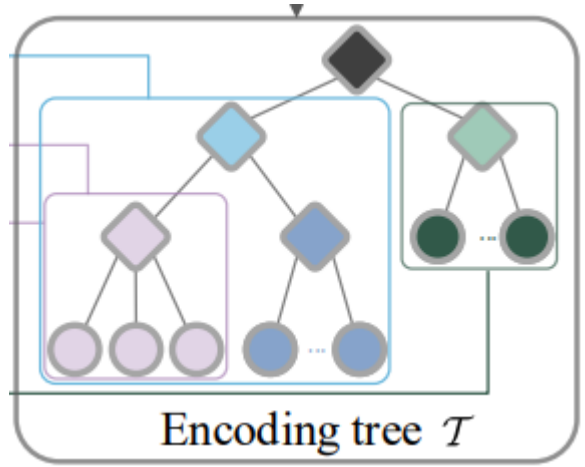
non-root node α parent node in T is denoted as α^-

its i-th children node $\alpha^{\langle i \rangle}$

non-leaf node α the number of children α is N

all vertex subset $T_{\alpha^{\langle i \rangle}}$ form a partition of T_α

$T_\alpha = \bigcup_{i=1}^{N} T_{\alpha^{\langle i \rangle}}$ and $\bigcap_{i=1}^{N} T_{\alpha^{\langle i \rangle}} = \varnothing$



K-level encoding tree.

One-dimensional Structural Entropy

$$H^1(G) = -\sum_{v \in V} \frac{d_v}{vol(G)} \log_2 \frac{d_v}{vol(G)}, \quad (1)$$

where d_v is the degree of vertex v, and $vol(G)$ is the sum of the degrees of all vertices in G.

High-dimensional Structural Entropy

$$H^K(G) = \min_{\forall \mathcal{T}: height(\mathcal{T}) \le K} \{H^{\mathcal{T}}(G)\}, \quad (2)$$

$$H^{\mathcal{T}}(G) = \sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} H^{\mathcal{T}}(G; \alpha) = -\sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} \frac{g_\alpha}{vol(G)} \log_2 \frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}, \quad (3)$$

where g_α is the sum weights of the cut edge set $[T_\alpha, T_\alpha / T_\lambda]$, i.e., all edges connecting vertices inside T_α with vertices outside T_α. $\mathcal{V}_\alpha$ is the sum of degrees of all vertices in T_α. $H^{\mathcal{T}}(G; \alpha)$ is the structural entropy of node α and $H^{\mathcal{T}}(G)$ is the structural entropy of $\mathcal{T}$. $H^K(G)$ is the K-dimensional structural entropy, with the optimal encoding tree of K-level .

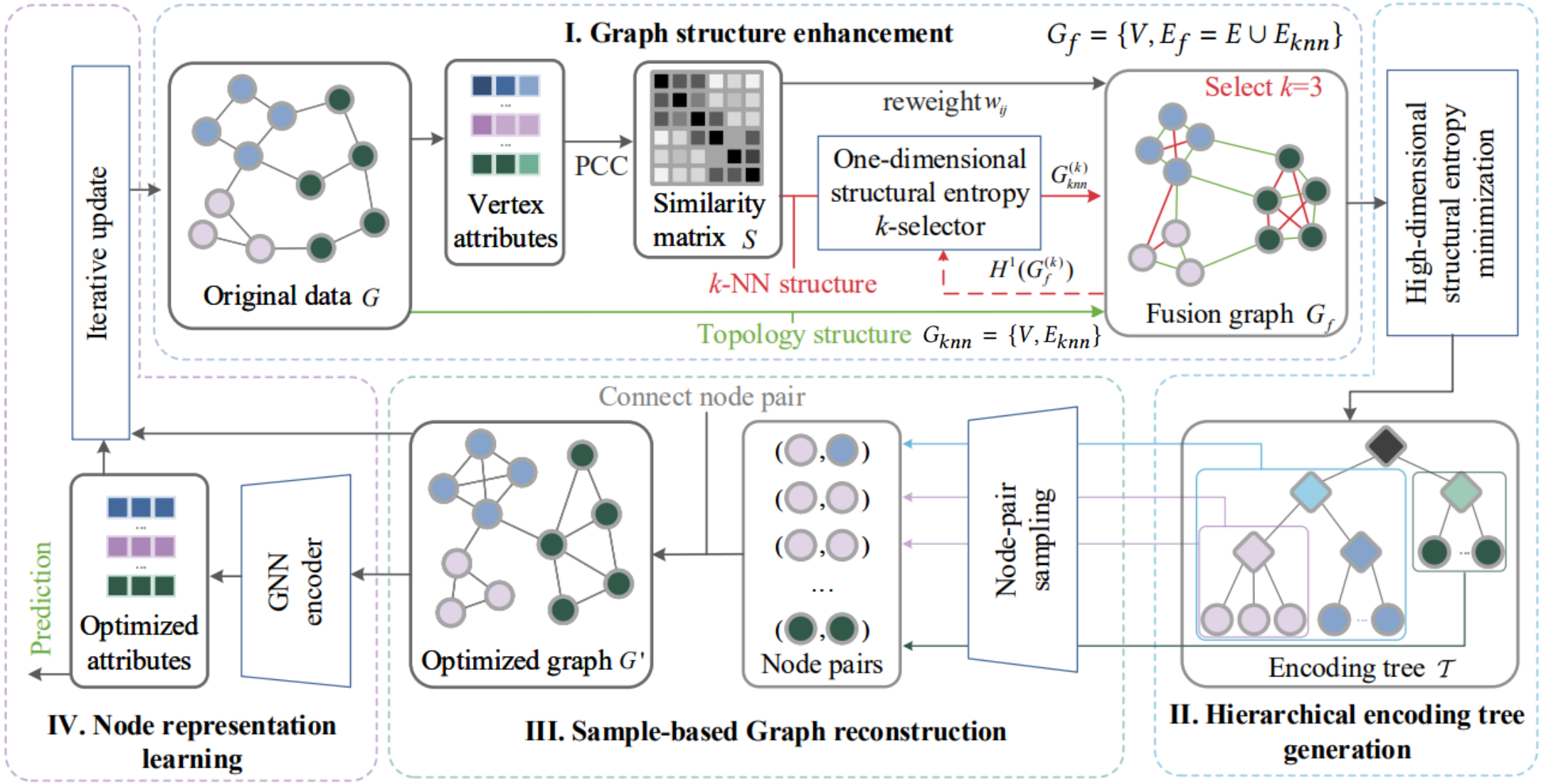
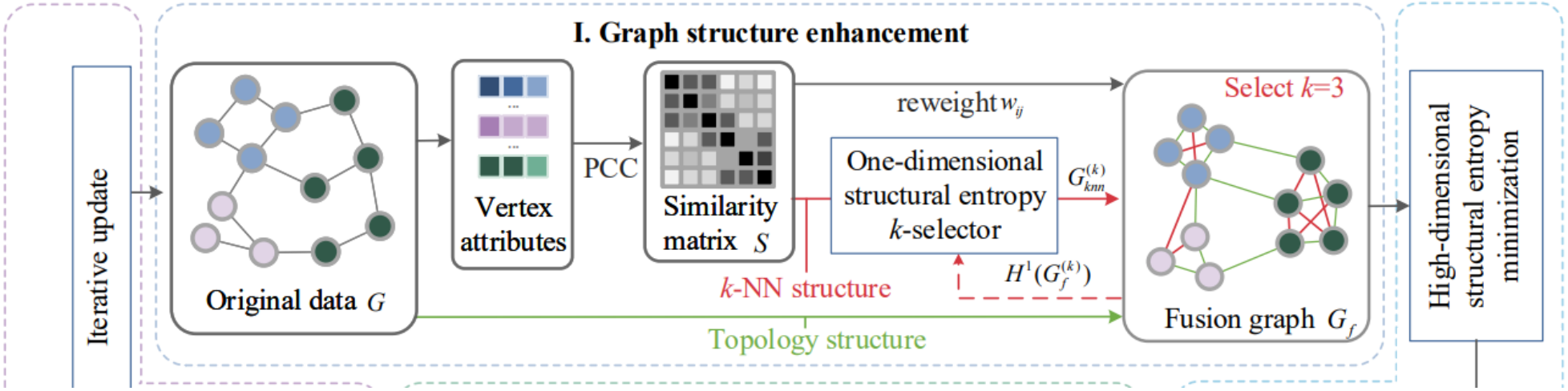


Figure 2: The overall architecture of SE-GSL.



Graph Structure Enhancement

$$S_{ij} = \mathrm{PCC}(x_i, x_j) = \frac{E((x_i - u_i)(x_j - u_j))}{\sigma_i \sigma_j}, \tag{4}$$

where x_i and $x_j \in \mathbb{R}^{1\times d}$ are the attribute vectors of vertices i and j, respectively. u_i and σ_i denote the mean value and variance of x_i, and $E(\cdot)$ is the dot product function. Based on S, we can intrinsically construct the k-NN graph $G_{knn} = \{V, E_{knn}\}$ where each edge in E_{knn} represents a vertex and its k nearest neighbors (e.g., the edges in red in Fig 2). We fuse G_{knn} and the original G to $G_f = \{V, E_f = E \cup E_{knn}\}$.

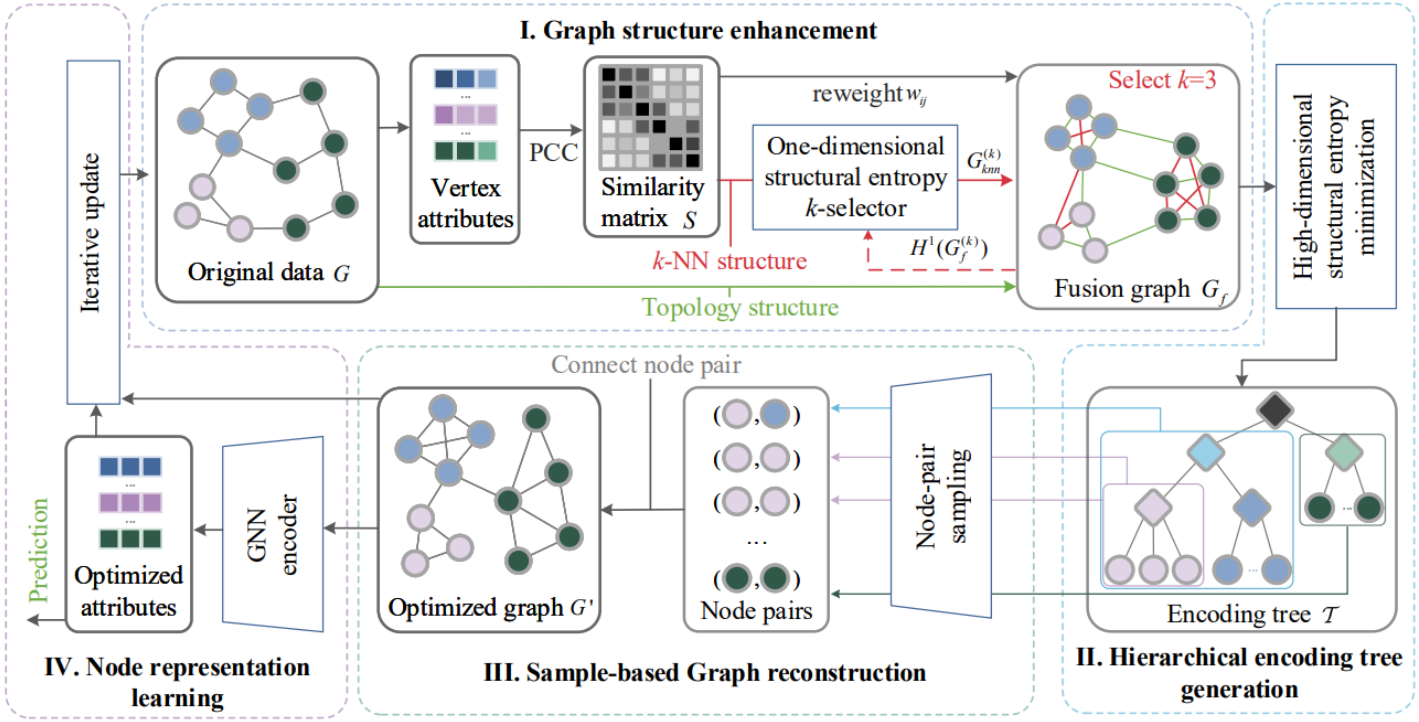
maximize the one-dimensional structural entropy $H^1(G_f)$ to guide the selection of k for larger encoding information. In practice, we gradually increase the integer parameter k, generate the corresponding $G_f^{(k)}$ and compute $H^1(G_f^{(k)})$.

k reaches a threshold k_m

the edge e_{ij} between v_i and v_j is reweighted as:

$$\omega_{ij} = S_{ij} + M, \quad M = \frac{1}{2|V|} \cdot \frac{1}{|E|} \sum_{1<i,j<n} S_{ij}, \tag{5}$$

where M is a modification factor that amplifies the trivial edge weights and thus makes the k-selector more sensitive to noises.

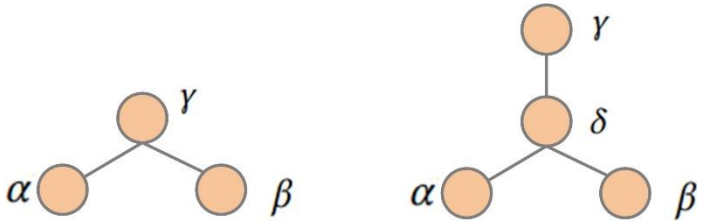


Hierarchical Encoding Tree Generation

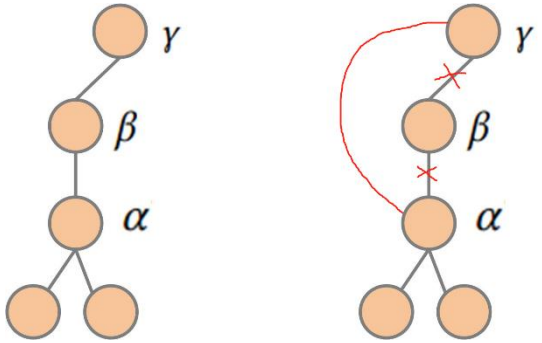
$$\mathcal{T}^* = \underset{\forall \mathcal{T}: height(\mathcal{T}) \le K}{\arg\min} (H^{\mathcal{T}}(G)). \quad (6)$$

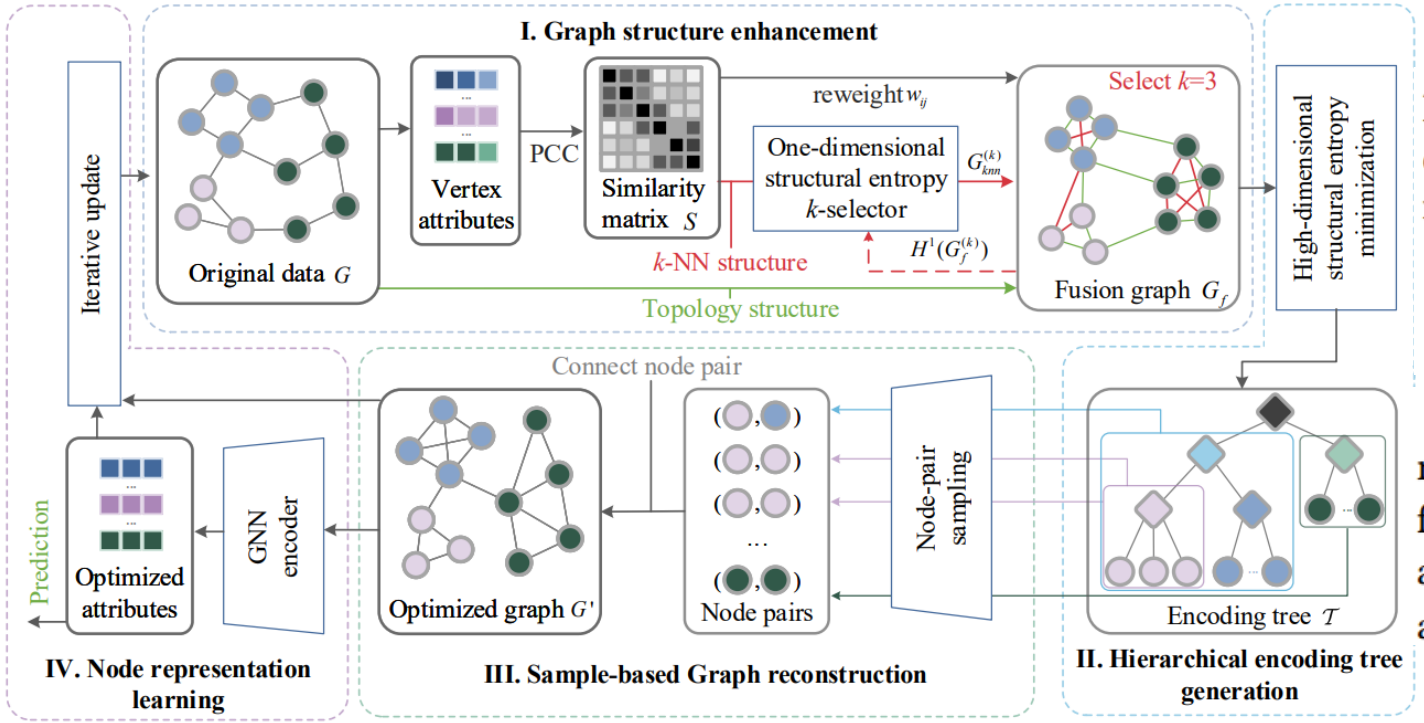
To address this optimization problem, we design a greedy-based heuristic algorithm to approximate $H^K(G)$. To assist the greedy heuristic, we define two basic operators:

Definition 1. Combining operator: Given an encoding tree $\mathcal{T}$ for $G = (V, E)$, let α and β be two nodes in $\mathcal{T}$ sharing the same parent γ. The combining operator $\mathrm{CB}_{\mathcal{T}}(\alpha, \beta)$ updates the encoding tree as: $\gamma \leftarrow \delta^-; \delta \leftarrow \alpha^-; \delta \leftarrow \beta^-$. A new node δ is inserted between γ and its children α, β.



Definition 2. Lifting operator: Given an encoding tree $\mathcal{T}$ for $G = (V, E)$, let α, β and γ be the nodes in $\mathcal{T}$, satisfying $\beta^- = \gamma$ and $\alpha^- = \beta$. The lifting operator $\mathrm{LF}_{\mathcal{T}}(\alpha, \beta)$ updates the encoding tree as: $\gamma \leftarrow \alpha^-$; IF :$T_\beta = \varnothing$, THEN :drop($\beta$). The subtree rooted at α is lifted by placing itself as γ's child. If no more children exist after lifting, β will be deleted from $\mathcal{T}$.





Sample-based Graph Reconstruction

Definition 3. Structural entropy of deduction: Let $\mathcal{T}$ be an encoding tree of G. We define the structural entropy of the deduction from non-leaf node λ to its descendant α as:

$$H^{\mathcal{T}}(G;(\lambda,\alpha]) = \sum_{\beta, T_\alpha \subseteq T_\beta \subset T_\lambda} H^{\mathcal{T}}(G;\beta). \tag{7}$$

Specifically, for a given $\mathcal{T}$, assume the node δ has a set of child nodes $\{\delta^{\langle 1\rangle}, \delta^{\langle 2\rangle}, \ldots, \delta^{\langle n\rangle}\}$. The probability of the child $\delta^{\langle i\rangle}$ is defined as: $P(\delta^{\langle i\rangle}) = \sigma_\delta(H^{\mathcal{T}}(G_f;(\lambda,\delta^{\langle i\rangle}])))$, where λ is the root of $\mathcal{T}$ and $\sigma_\delta(\cdot)$ represents a distribution function. Take softmax function as an example, the probability of $\delta^{\langle i\rangle}$ can be calculated as:

$$P(\delta^{\langle i\rangle}) = \frac{\exp(H^{\mathcal{T}}(G_f;(\lambda,\delta^{\langle i\rangle}])))}{\sum_{j=1}^{n}\exp(H^{\mathcal{T}}(G_f;(\lambda,\delta^{\langle j\rangle}])))}. \tag{8}$$

(1) For the encoding tree (or subtree) with root node δ, two different child nodes $\delta^{\langle i\rangle}$ and $\delta^{\langle j\rangle}$ are selected by sampling according to $P(\delta^{\langle i\rangle})$ and $P(\delta^{\langle j\rangle})$. Let $\delta_1 \leftarrow \delta^{\langle i\rangle}$ and $\delta_2 \leftarrow \delta^{\langle j\rangle}$

(2) If δ_1 is a non-leaf node, we perform sampling once on the subtree rooted at δ_1 to get $\delta_1^{\langle i\rangle}$, then update $\delta_1 \leftarrow \delta_1^{\langle i\rangle}$. The same is operated on δ_2.

(3) After recursively performing step (2), we sample two leaf nodes δ_1 and δ_2, while adding the edge connecting vertex $v_1 = T_{\delta_1}$ and $v2 = T_{\delta_2}$ into the edge set E' of graph G'.

For each subtree rooted at δ, we conduct independent samplings for $\theta \times n$ times, where n is the number of δ's children,

θ is a hyperparameter that positively correlated with the density of reconstructed graph.

Table 1: Classification Accuracy (%) comparison, with improvement range of SE-GSL against the baselines. The best results are bolded and the second-best are underlined. Green denotes the outperformance percentage, while yellow denotes underperformance.

Method	Cora	Citeseer	Pubmed	PT	TW	Actor	Cornell	Texas	Wisconsin
GCN	$87.26_{\pm 0.63}$	$76.22_{\pm 0.71}$	$87.46_{\pm 0.12}$	$67.62_{\pm 0.21}$	$62.46_{\pm 1.94}$	$27.65_{\pm 0.55}$	$49.19_{\pm 1.80}$	$57.30_{\pm 2.86}$	$48.57_{\pm 4.08}$
GAT	$87.52_{\pm 0.69}$	$76.04_{\pm 0.78}$	$86.61_{\pm 0.15}$	$68.76_{\pm 1.01}$	$61.68_{\pm 1.20}$	$27.77_{\pm 0.59}$	$57.09_{\pm 6.32}$	$58.10_{\pm 4.14}$	$51.34_{\pm 4.78}$
GCNII	$87.57_{\pm 0.87}$	$75.47_{\pm 1.01}$	$\underline{88.64_{\pm 0.23}}$	$68.93_{\pm 0.93}$	$65.17_{\pm 0.47}$	$30.66_{\pm 0.66}$	$58.76_{\pm 7.11}$	$55.36_{\pm 6.45}$	$51.96_{\pm 4.36}$
Grand	$\mathbf{87.93}_{\pm 0.71}$	$77.59_{\pm 0.85}$	$86.14_{\pm 0.98}$	$69.80_{\pm 0.75}$	$\underline{66.79_{\pm 0.22}}$	$29.80_{\pm 0.60}$	$57.21_{\pm 2.48}$	$56.56_{\pm 1.53}$	$52.94_{\pm 3.36}$
Mixhop	$85.71_{\pm 0.85}$	$75.94_{\pm 1.00}$	$87.31_{\pm 0.44}$	$69.48_{\pm 0.30}$	$66.34_{\pm 0.22}$	$33.72_{\pm 0.76}$	$64.47_{\pm 4.78}$	$63.16_{\pm 6.28}$	$72.12_{\pm 3.34}$
Dropedge	$86.32_{\pm 1.09}$	$76.12_{\pm 1.32}$	$87.58_{\pm 0.34}$	$68.49_{\pm 0.91}$	$65.24_{\pm 1.45}$	$30.10_{\pm 0.71}$	$58.94_{\pm 5.95}$	$59.20_{\pm 5.43}$	$60.45_{\pm 4.48}$
Geom-GCN-P	84.93	75.14	88.09	-	-	31.63	60.81	67.57	64.12
Geom-GCN-S	85.27	74.71	84.75	-	-	30.30	55.68	59.73	56.67
GDC	$87.17_{\pm 0.36}$	$76.13_{\pm 0.53}$	$88.08_{\pm 0.16}$	$66.14_{\pm 0.54}$	$64.14_{\pm 1.40}$	$28.74_{\pm 0.76}$	$59.46_{\pm 4.35}$	$56.42_{\pm 3.99}$	$48.30_{\pm 4.29}$
GEN	$\underline{87.84_{\pm 0.69}}$	$\mathbf{78.77}_{\pm 0.88}$	$86.13_{\pm 0.41}$	$\underline{71.62_{\pm 0.78}}$	$65.16_{\pm 0.77}$	$\mathbf{36.69}_{\pm 1.02}$	$65.57_{\pm 6.74}$	$73.38_{\pm 6.65}$	$54.90_{\pm 4.73}$
H_2GCN-2	$87.81_{\pm 1.35}$	$76.88_{\pm 1.77}$	$\mathbf{89.59}_{\pm 0.33}$	$68.15_{\pm 0.30}$	$63.33_{\pm 0.77}$	$35.62_{\pm 1.30}$	$\mathbf{82.16}_{\pm 6.00}$	$\underline{82.16_{\pm 5.28}}$	$\underline{85.88_{\pm 4.22}}$
SE-GSL	$\mathbf{87.93}_{\pm 1.24}$	$\underline{77.63_{\pm 1.65}}$	$88.16_{\pm 0.76}$	$\mathbf{71.91}_{\pm 0.66}$	$\mathbf{66.99}_{\pm 0.93}$	$\underline{36.34_{\pm 2.07}}$	$\underline{75.21_{\pm 5.54}}$	$\mathbf{82.49}_{\pm 4.80}$	$\mathbf{86.27}_{\pm 4.32}$
Improvement	0.00~3.00	-1.14~2.92	-1.43~3.41	0.29~5.77	0.20~5.31	-0.35~8.69	-6.95~26.02	0.33~27.13	0.39~37.97

Experiments

Table 2: Classification accuracy(%) of SE-GSL and corresponding backbones. Wisc. is short for Wisconsin.

Method	Actor	TW	Texas	Wisc.	Improvement
SE-GSL$_{GCN}$	35.03	66.88	75.68	79.61	↑ 5.20~31.04
SE-GSL$_{SAGE}$	36.20	66.92	**82.49**	**86.27**	↑ 0.25~6.79
SE-GSL$_{GAT}$	32.46	63.57	74.59	78.82	↑ 4.69~27.48
SE-GSL$_{APPNP}$	**36.34**	**66.99**	81.28	83.14	↑ 2.01~12.16

Table 3: The k selection for each iteration in structural optimization. Bolds represent the k selection when the accuracy reaches maximum.

Iteration	1	2	3	4	5	6	7	8	9
Cora	22	**22**	19	22	21	22	20	21	20
Actor	23	15	15	15	14	15	14	**14**	15
TW	50	16	16	**17**	15	17	27	16	16
Wisconsin	21	16	**11**	16	14	13	16	13	11
Texas	21	13	13	**13**	13	10	14	10	14

Table 4: Glossary of Notations.

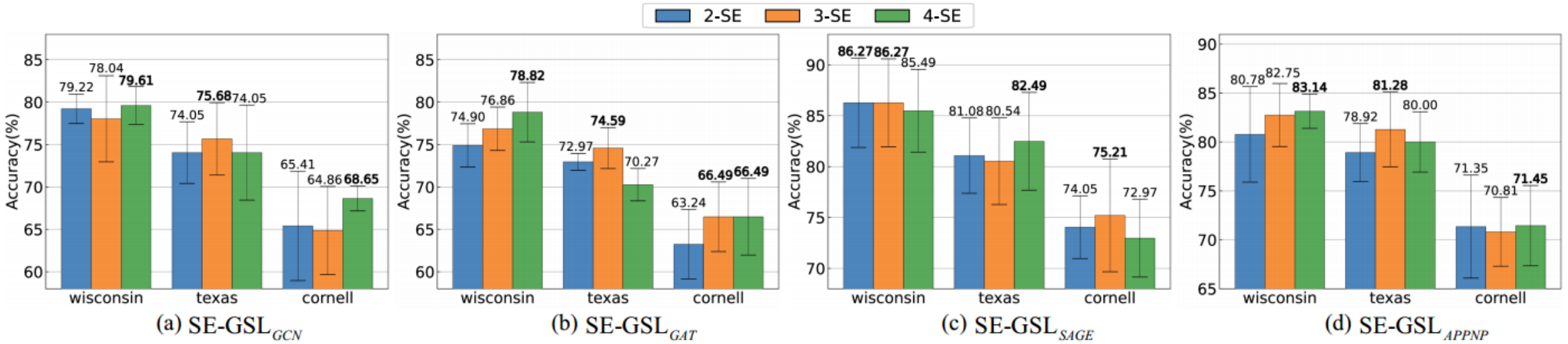
Notation	Description
$G; A; S$	Graph; Adjacency matrix; Similarity matrix.
$v; e; x$	Vertex; Edge; Vertex attribute.
$V; E; X$	Vertex set; Edge set; Attribute set.
$\|V\|; \|E\|$	The number of vertices and edges.
$\mathcal{P}; P_i$	The partition of V; A community.
$D; d(v_i)$	The degree matrix; The degree of vertex v_i.
e_{ij}	The edge between v_i and v_j.
w_{ij}	The weight of edge e_{ij}.
$vol(G)$	The volume of graph G, i.e., degree sum in G.
$G_{knn}^{(k)}$	The k-NN graph with parameter k.
G_f	Fusion graph.
$G_f^{(k)}$	The fusion graph with parameter k.
$\mathcal{T}$	Encoding tree.
$\mathcal{T}^*$	The optimal encoding tree.
λ	The root node of an encoding tree.
α	A non-root node of an encoding tree.
α^-	The parent node of α.
$\alpha^{\langle i \rangle}$	the i-th child of α.
T_λ	The label of λ, i.e., node set V.
T_α	The label of α, i.e., a subset of V.
$\mathcal{V}_\alpha$	Volume of graph G.
g_a	the sum weights of cut edge set $[T_\alpha, T_\alpha/T_\lambda]$.
$N(\mathcal{T})$	The number of non-root node in $\mathcal{T}$.
$H^{\mathcal{T}}(G)$	Structural entropy of G under $\mathcal{T}$.
$H^K(G)$	K-dimensional structural entropy.
$H^1(G)$	One-dimensional structural entropy.
$H^{\mathcal{T}}(G; \alpha)$	Structural entropy of node α in $\mathcal{T}$.
$H^{\mathcal{T}}(G; (\lambda, \alpha])$	Structural entropy of a deduction from λ to α.

Table 5: Statistics of benchmark datasets.

Dataset	Nodes	Edges	Classes	Features	homophily
Cora	2708	5429	7	1433	0.83
Citeseer	3327	4732	6	3703	0.71
Pubmed	19717	44338	3	500	0.79
PT	1912	31299	2	3169	0.59
TW	2772	63462	2	3169	0.55
Actor	7600	33544	5	931	0.24
Cornell	183	295	5	1703	0.30
Texas	183	309	5	1703	0.11
Wisconsin	251	499	5	1703	0.21

Table 6: Comparison of training time(hr.) of achieving the best performance based on GPU.

Method	Cora	Citeseer	Pubmed	PT	TW	Actor	Cornell	Texas	Wisconsin
SE-GSL$_{GCN}$	0.071	0.213	4.574	0.178	0.374	1.482	0.006	0.008	0.009
SE-GSL$_{SAGE}$	0.074	0.076	4.852	0.169	0.214	0.817	0.006	0.007	0.009
SE-GSL$_{GAT}$	0.071	0.180	4.602	0.172	0.329	1.273	0.006	0.008	0.009
SE-GSL$_{APPNP}$	0.073	0.215	4.854	0.138	0.379	1.367	0.010	0.011	0.013



(a) SE-GSL$_{GCN}$ (b) SE-GSL$_{GAT}$ (c) SE-GSL$_{SAGE}$ (d) SE-GSL$_{APPNP}$

Figure 3: Results of SE-GSL with different encoding tree heights.

Experiments

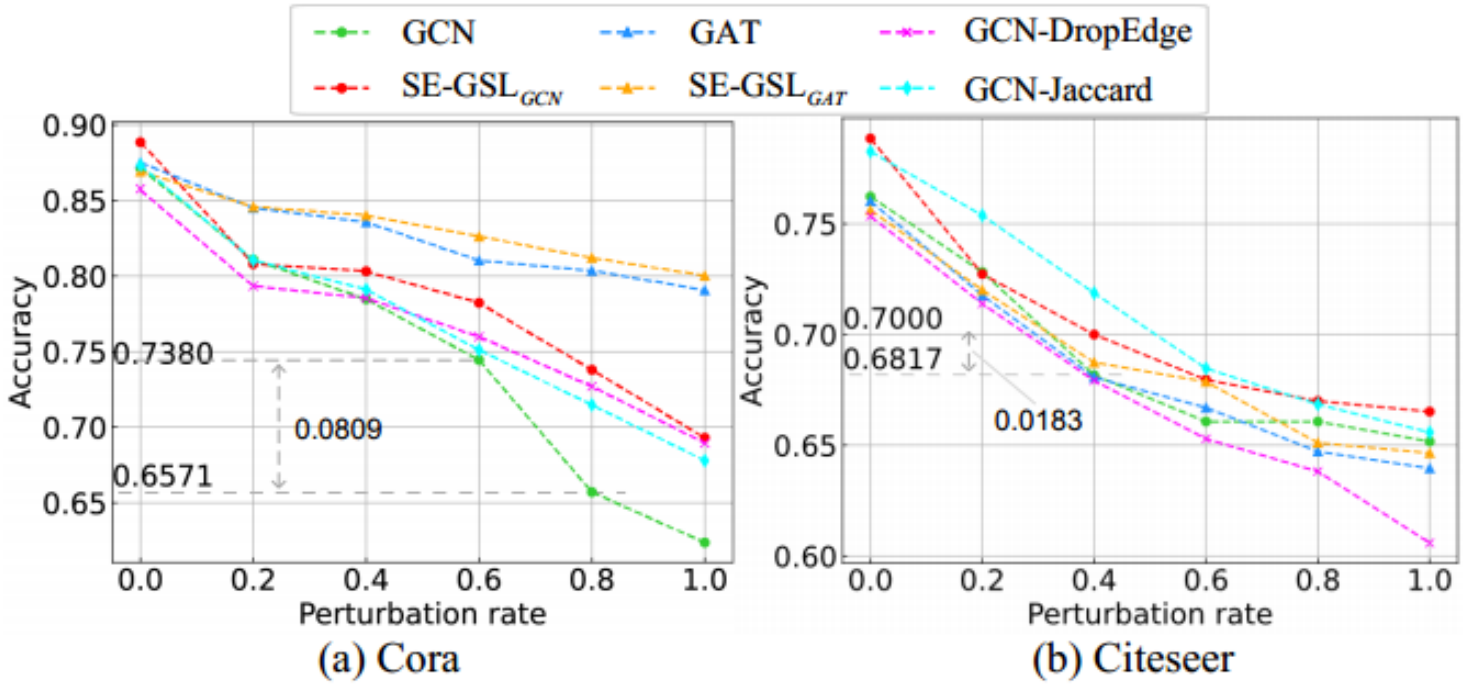


Figure 4: Robustness of SE-GSL against random noises.

Experiments

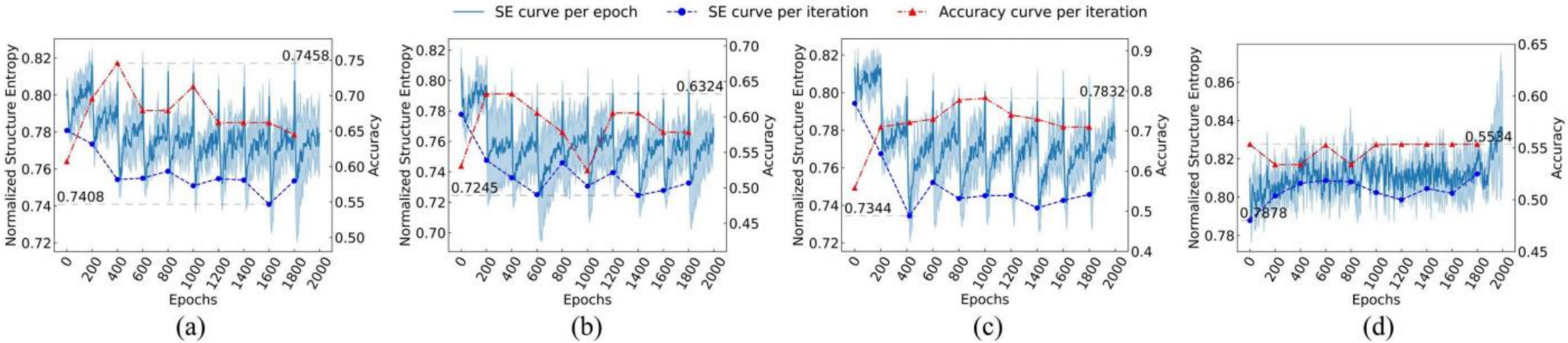


Figure 5: The normalized structural entropy changes during the training of SE-GSL$_{GAT}$ with 2-dimensional structural entropy on (a) Texas, (b) Cornell, and (c) Wisconsin. The structure is iterated every 200 epochs. By comparison, (d) shows the entropy changes on Wisconsin without the graph reconstruction strategy.

Experiments

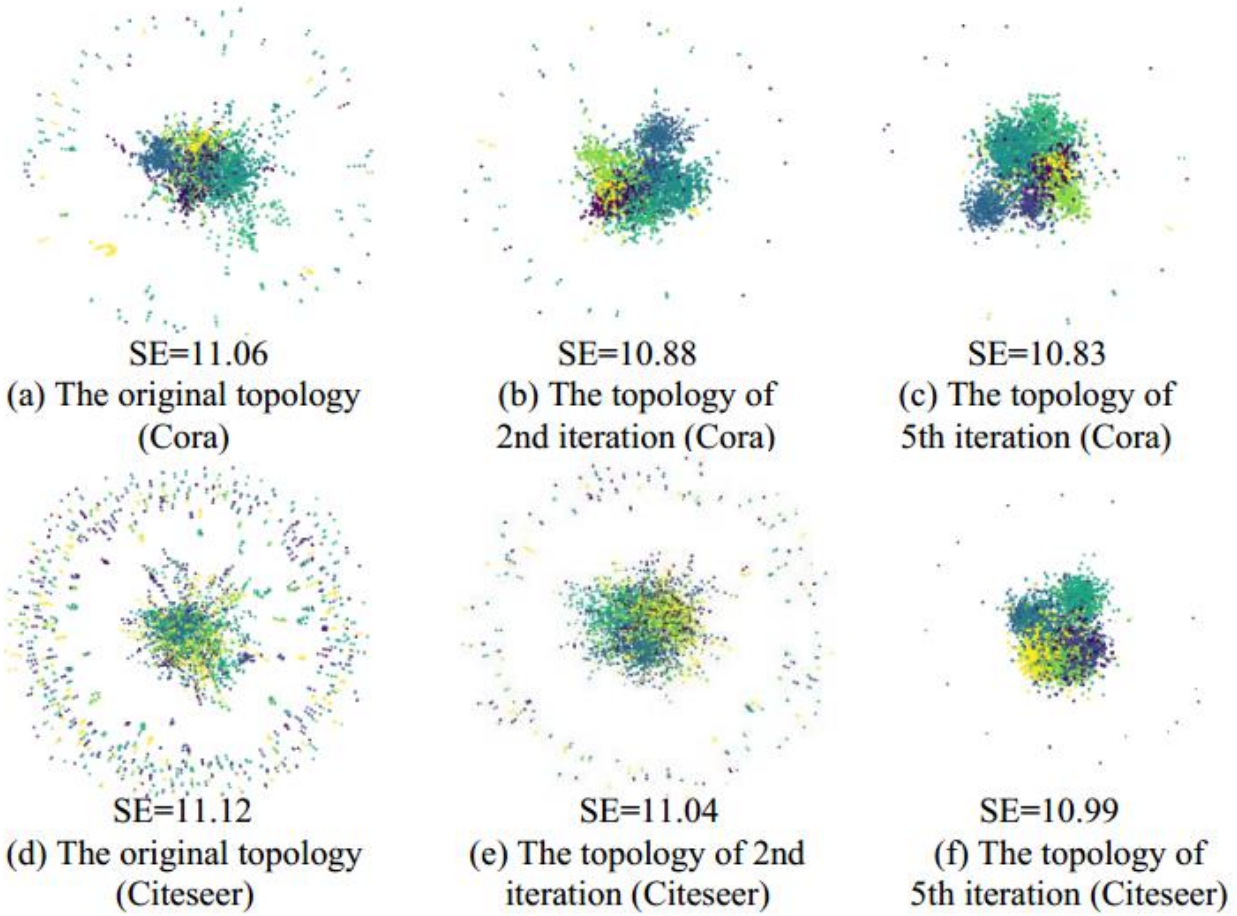


Figure 6: The visualized evolution of the graph structure on Cora (a,b,c) and Citeseer (d,e,f). The corresponding Structural Entropy (SE) is also shown.

Thanks