Structural Entropy Guided Graph Hierarchical Pooling

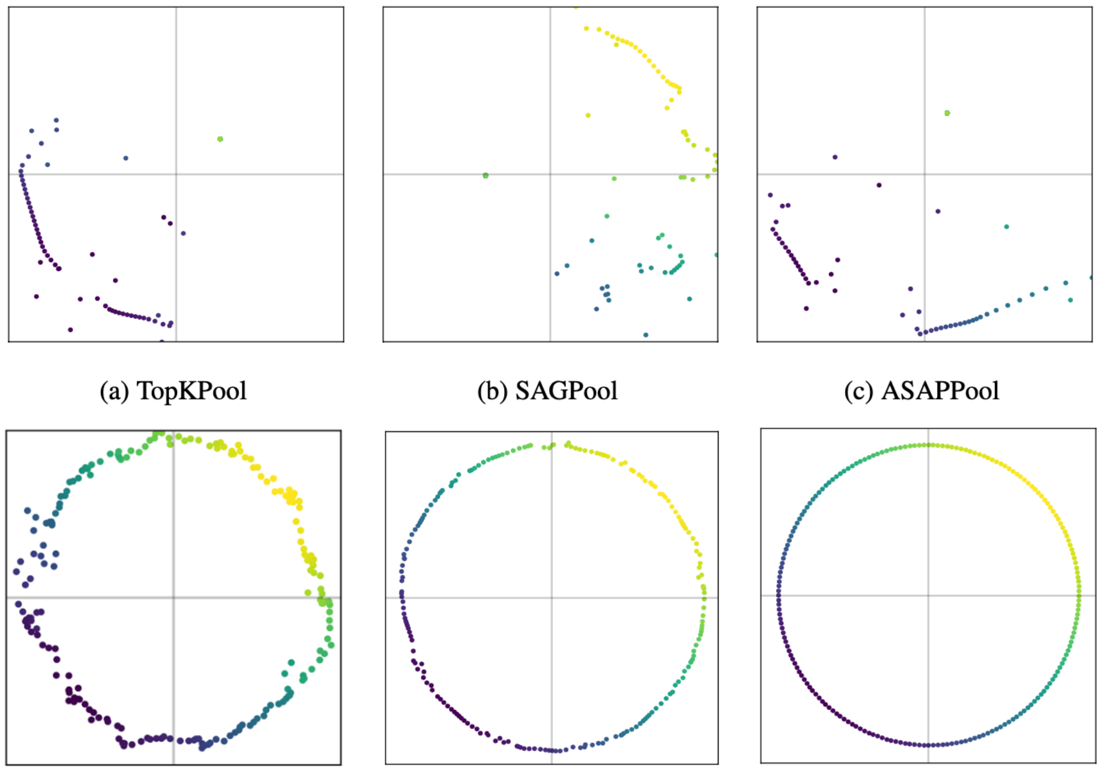
Junran Wu[1*], Xueyuan Chen[1*], Ke Xu[1], Shangzhe Li[12]

[1]State Key Lab of Software Development Environment, Beihang University

[2]School of Mathematical Science, Beihang University

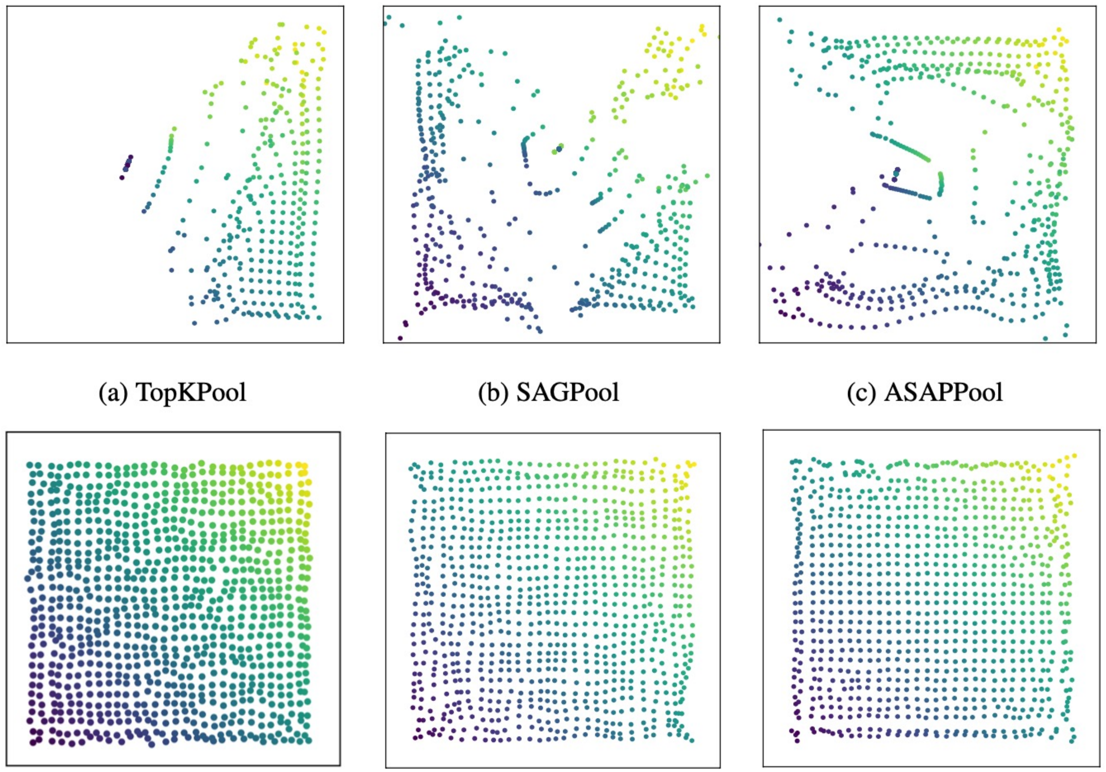
北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University

Motivation



(a) TopKPool (b) SAGPool (c) ASAPPool

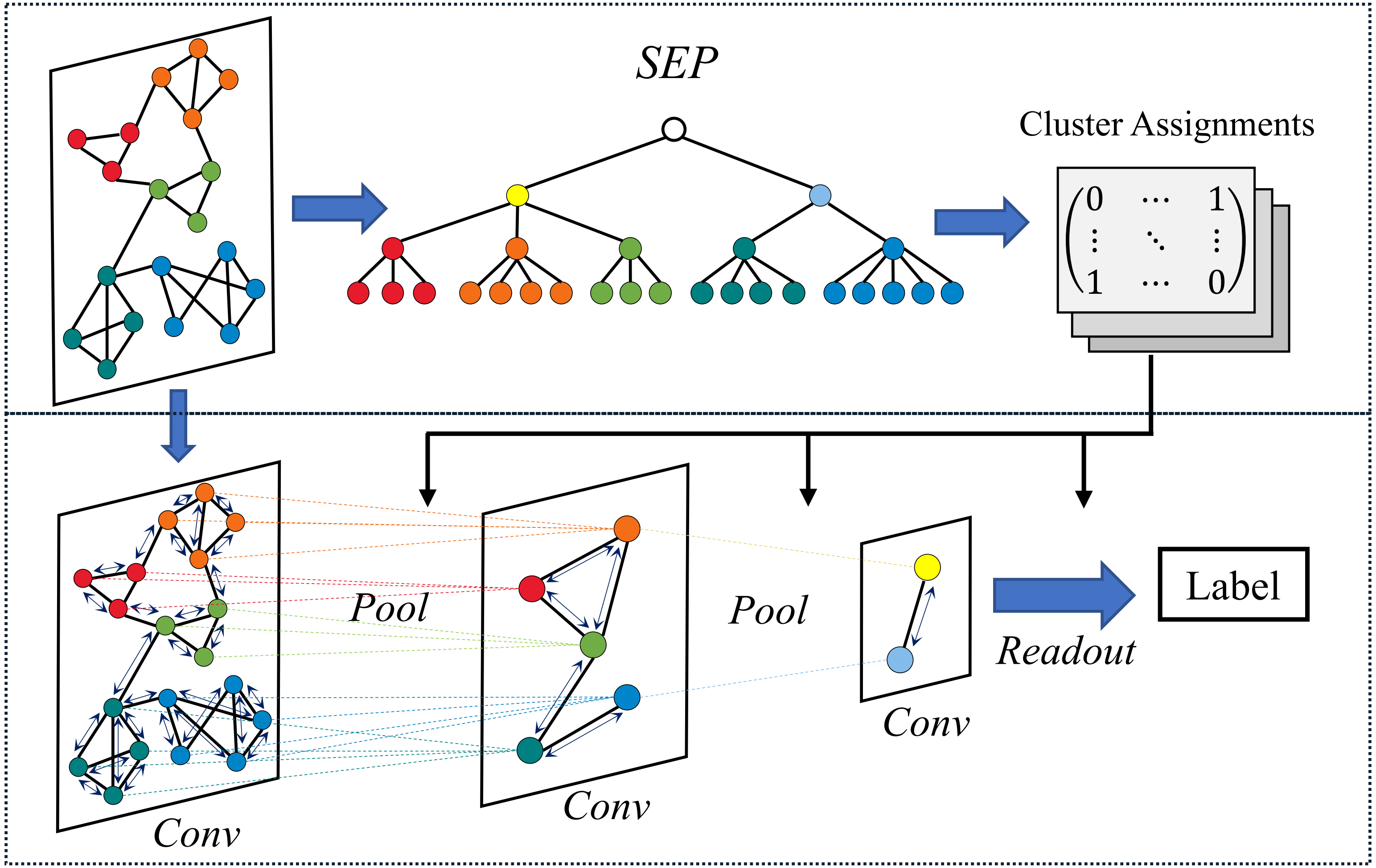
(d) DiffPool (e) minCutPool (f) SEP



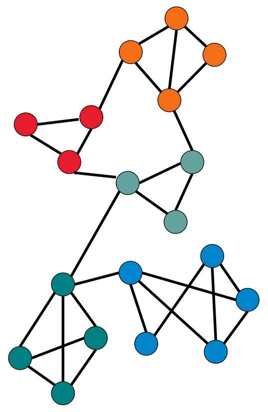
(a) TopKPool (b) SAGPool (c) ASAPPool

(d) DiffPool (e) minCutPool (f) SEP

Framework Overview

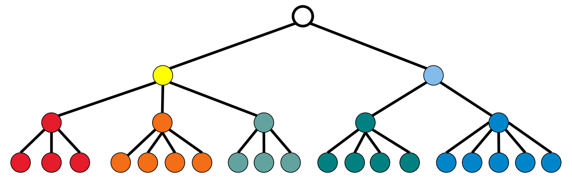


Structural Entropy



$$\mathcal{H}^T(G) = -\sum_{v_t \in T} \frac{g_{v_t}}{vol(V)} \log \frac{vol(v_t)}{vol(v_t^+)}$$

$$\mathcal{H}(G) = \min_T \{\mathcal{H}^T(G)\}$$



k-Dimensional Structural Entropy

$$\mathcal{H}^{(k)}(G) = \min_{T:\text{height}(T) \le k} \{\mathcal{H}^T(G)\}$$

"Structural information and dynamical complexity of networks."
IEEE Transactions on Information Theory. Li, Angsheng, and Yicheng Pan.

Structural Entropy Minimization

Definition 3.1. Let T be any coding tree for graph $G = (\mathcal{V}, \mathcal{E})$, v_r is the root node of T and $\mathcal{V}$ are the leaf nodes of T. Given any two nodes (v_i, v_j) in T, in which $v_i \in v_r.children$ and $v_j \in v_r.children$.

Define a function $\text{MERGE}_T(v_i, v_j)$ for T to insert a new node v_ε between v_r and (v_i, v_j):

$$v_\varepsilon.children \leftarrow v_i; \tag{5}$$

$$v_\varepsilon.children \leftarrow v_j; \tag{6}$$

$$v_r.children \leftarrow v_\varepsilon; \tag{7}$$

Definition 3.2. Following the setting in Definition 3.1, given any two nodes (v_i, v_j), in which $v_i \in v_j.children$.

Define a function $\text{REMOVE}_T(v_i)$ for T to remove v_i from T and merge $v_i.children$ to $v_j.chileren$:

$$v_j.children \leftarrow v_i.children; \tag{8}$$

Definition 3.3. Following the setting in Definition 3.1, given any two nodes (v_i, v_j), in which $v_i \in v_j.children$ and $|Heigth(v_j) - Height(v_i)| > 1$.

Define a function $\text{FILL}(v_i, v_j)$ for T to insert a new node v_ε between v_i and v_j:

$$v_\varepsilon.children \leftarrow v_i; \tag{9}$$

$$v_j.children \leftarrow v_\varepsilon; \tag{10}$$

Structural Entropy Minimization

Given, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$,

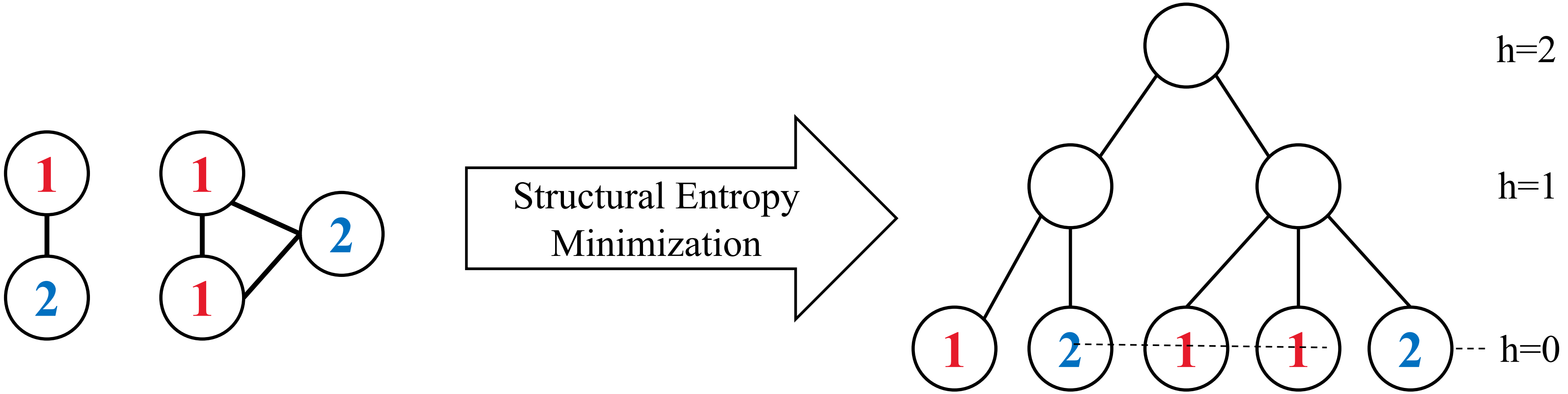
$\boldsymbol{CT} = \text{SE}_{\boldsymbol{min}}(\boldsymbol{G})$, where $\boldsymbol{CT} = (\boldsymbol{V_T}, \boldsymbol{E_T})$, and

$\boldsymbol{V_T} = (\boldsymbol{V_T^0}, \boldsymbol{V_T^1}, \dots, \boldsymbol{V_T^k}), \boldsymbol{V_T^0} = \boldsymbol{V}$

Algorithm 1 Coding tree with height k via structural entropy minimization

Input: a graph $G = (\mathcal{V}, \mathcal{E})$, a positive integer $k > 1$
Output: a coding tree T with height k

```
Generate a coding tree T with a root node v_r and all nodes in V as leaf nodes;
// Stage 1: Bottom to top construction;
while |v_r.children| > 2 do
    Select v_i and v_j from v_r.children, conditioned on
    argmax_(v_i,v_j){H^T(G) − H^(T_MERGE(v_i,v_j))(G)};
    MERGE(v_i, v_j);
end while
// Stage 2: Compress T to the certain height k;
while Height(T) > k do
    Select v_i from T, conditioned on
    argmin_(v_i){H^(T_REMOVE(v_i))(G) − H^T(G)|
                  v_i ≠ v_r & v_i ∉ V};
    REMOVE(v_i);
end while
// Stage 3: Fill T to avoid cross-layer links;
for v_i ∈ T do
    if |Height(v_i.parent) − Height(v_i)| > 1 then
        FILL(v_i, v_i.parent);
    end if
end for
return T;
```



北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University

Graph / Node Classification

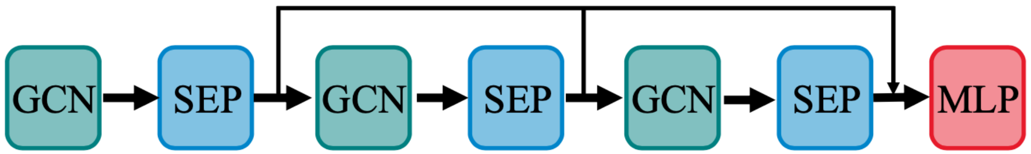


Figure 2: **The SEP-G architecture for graph classification.** Following the design of previous works in hierarchical pooling, the architecture is comprised of three GCN layers and each is followed by corresponding SEP layer.

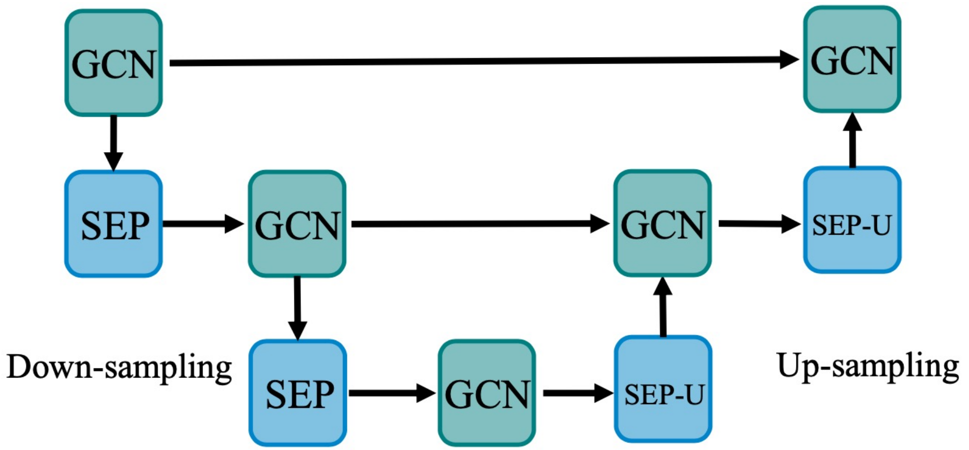


Figure 3: **The SEP-N architecture for node classification.** There are two encoder and two decoder blocks and each block is composed of a GCN layer and a pooling (unpooling) layer. Skip connection links the same-level encoder and decoder to enhance spatial feature transmission.

Evaluation-Graph Classification

Table 1: **Graph classification accuracies on seven benchmarks (%).** The shown accuracies are mean and standard deviation over 10 different runs. We highlight the best results.

		Social Network			Bioinformatics			
		IMDB-BINARY	IMDB-MULTI	COLLAB	MUTAG	PROTEINS	D&D	NCI1
# Graphs		1,000	1,500	5,000	188	1,113	1,178	4,110
# Classes		2	3	3	2	2	2	2
Avg. # Nodes		19.8	13.0	74.5	17.9	39.1	284.3	29.8
Backbones	GCN	73.26±0.46	50.39±0.41	80.59±0.27	69.50±1.78	73.24±0.73	72.05±0.55	76.29±1.79
	GIN	72.78±0.86	48.13±1.36	78.19±0.63	81.39±1.53	71.46±1.66	70.79±1.17	**80.00±1.40**
Global Pooling	Set2Set	72.90±0.75	50.19±0.39	79.55±0.39	69.89±1.94	73.27±0.85	71.94±0.56	68.55±1.92
	SortPool	72.12±1.12	48.18±0.83	77.87±0.47	71.94±3.55	73.17±0.88	75.58±0.72	73.82±1.96
	SAGPool(G)	72.16±0.88	49.47±0.56	78.85±0.56	76.78±2.12	72.02±1.08	71.54±0.91	74.18±1.20
	StructPool	72.06±0.64	50.23±0.53	77.27±0.51	79.50±0.75	75.16±0.86	78.45±0.40	78.64±1.53
	GMT	73.48±0.76	50.66±0.82	80.74±0.54	83.44±1.33	75.09±0.59	**78.72±0.59**	76.35±2.62
Hierarchical Pooling	DiffPool	73.14±0.70	51.31±0.72	78.68±0.43	79.22±1.02	73.03±1.00	77.56±0.41	62.32±1.90
	SAGPool(H)	72.55±1.28	50.23±0.44	78.03±0.31	73.67±4.28	71.56±1.49	74.72±0.82	67.45±1.11
	TopKPool	71.58±0.95	48.59±0.72	77.58±0.85	67.61±3.36	70.48±1.01	73.63±0.55	67.02±2.25
	ASAP	72.81±0.50	50.78±0.75	78.64±0.5	77.83±1.49	73.92±0.63	76.58±1.04	71.48±0.42
	MinCutPool	72.65±0.75	51.04±0.70	80.87±0.34	79.17±1.64	74.72±0.48	78.22±0.54	74.25±0.86
	SEP-G	**74.12±0.56**	**51.53±0.65**	**81.28±0.15**	**85.56±1.09**	**76.42±0.39**	77.98±0.57	78.35±0.33

Evaluation-Node Classification

Table 3: **Node classification accuracies on Cora, Citeseer, and Pubmed (%).** We highlight our results and those that are significantly higher than all other methods.

	Cora	Citeseer	Pubmed
# Nodes	2,708	3,327	19,717
# Edges	5,429	4,732	44,338
# Features	1,433	3,703	4,500
# Classes	7	6	3
GCN	81.4±0.4	70.9±0.5	79.0±0.4
GAT	83.3±0.7	72.6±0.6	78.5±0.3
GIN	77.6±1.1	66.1±0.9	77.0±1.2
FastGCN	79.8±0.3	68.8±0.6	77.4±0.3
APPNP	83.3±0.5	71.7±0.6	80.1±0.2
MixHop	81.8±0.6	71.4±0.8	80.0±1.1
DGI	82.5±0.7	71.6±0.7	78.4±0.7
SGC	81.0±0.03	71.9±0.11	78.9±0.01
S^2GC	83.5±0.02	**73.6±0.09**	80.2±0.02
GCNII	**85.5±0.5**	73.4±0.6	**80.3**±0.4
g-U-Nets	84.4±0.6	73.2±0.5	79.6±0.2
SEP-N	**84.8±0.4**	**72.9±0.7**	**80.2±0.8**

	Model (#Convs)				
	S^2GC(4)	S^2GC(8)	GCNII(4)	GCNII(8)	SEP-N(5)
Cora	79.8	82.2	82.6	84.2	**84.8**
Citeseer	72.6	72.7	68.9	70.6	**72.9**
Pubmed	79.2	79.7	78.8	79.3	**80.2**

	Cora		Citeseer		Pubmed	
Depth	g-U-Nets	SEP-N	g-U-Nets	SEP-N	g-U-Nets	SEP-N
1	–	84.3±0.6	–	**73.3±0.6**	–	78.9±0.6
2	82.6±0.6	**84.8±0.4**	71.8±0.5	72.9±0.7	79.1±0.3	**80.2±0.8**
3	83.8±0.7	84.5±0.3	72.7±0.7	72.1±0.6	79.4±0.4	79.5±0.5
4	**84.4±0.6**	83.6±0.6	**73.2±0.5**	72.1±0.2	**79.6±0.2**	78.5±0.3
5	84.1±0.5	83.9±0.5	72.8±0.6	72.4±0.6	79.5±0.3	79.8±0.7

Contributions

- We uncover two crucial issues in previous hierarchical pooling works that hinder the development of GNNs, including the local structure damage and suboptimal problem because of the fixed compression quota and stepwise pooling design.

- Through the introduction of the structural information theory, we present a novel hierarchical pooling approach, termed SEP, to address the unveiled issues.

- We extensively validate SEP on graph reconstruction, graph classification, and node classification tasks, in which outperformances are observed in comparison the SOTA hierarchical pooling methods.

Thanks for watching

Code will be available at:

https://github.com/Wu-Junran/SEP

北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University