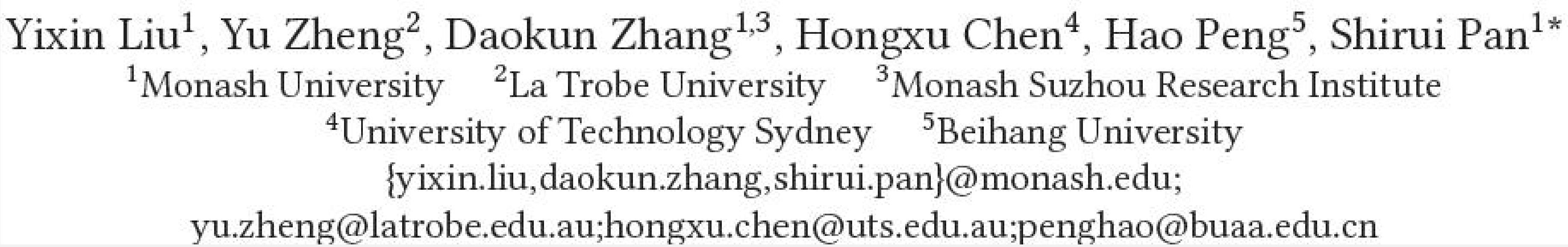
# Towards Unsupervised Deep Graph Structure Learning

Code:https://g



Yixin Liu[1], Yu Zheng[2], Daokun Zhang[1,3], Hongxu Chen[4], Hao Peng[5], Shirui Pan[1*]
[1]Monash University [2]La Trobe University [3]Monash Suzhou Research Institute
[4]University of Technology Sydney [5]Beihang University
{yixin.liu,daokun.zhang,shirui.pan}@monash.edu;
yu.zheng@latrobe.edu.au;hongxu.chen@uts.edu.au;penghao@buaa.edu.cn

Code : https://github.com/GRAND-Lab/SUBLIME

WWW_2022

gesis
Leibniz-Institut
für Sozialwissenschaften

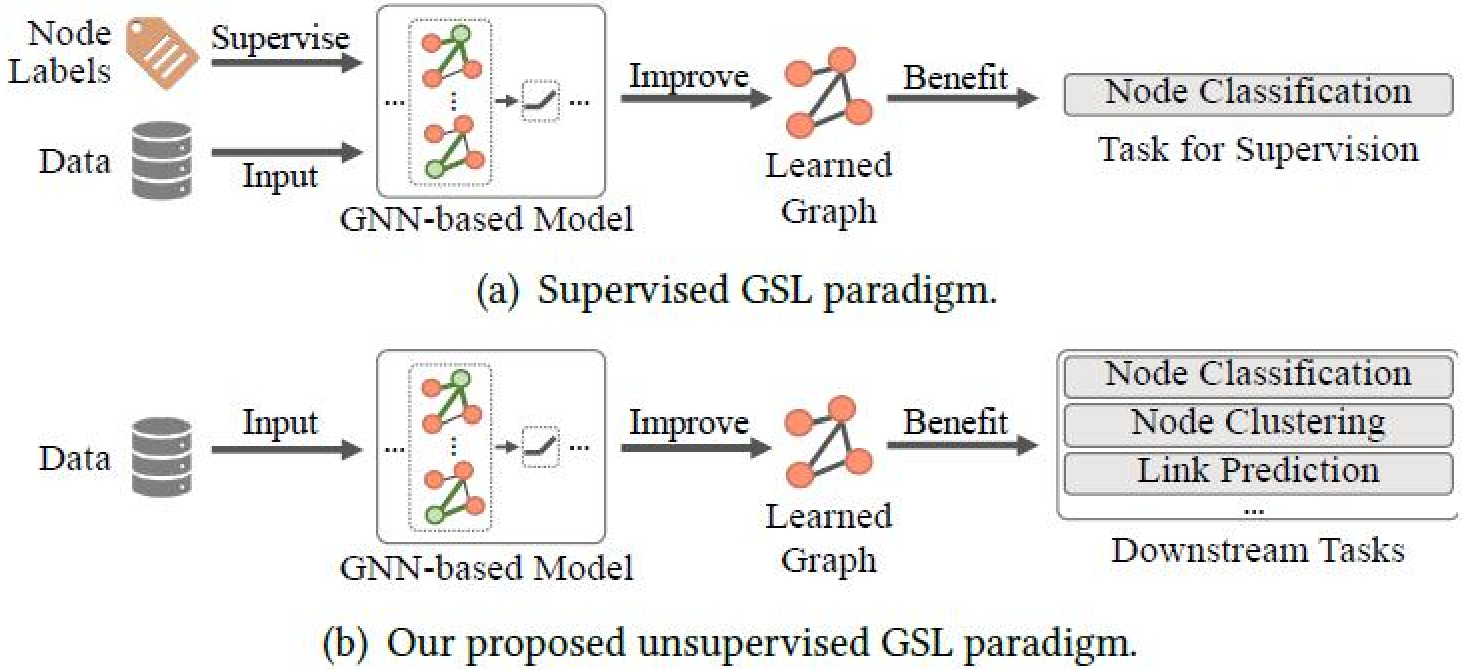
Reported by Xinsheng Wang

1.Introduction

2.Method

3.Experiments

# Introduction



(a) Supervised GSL paradigm.

(b) Our proposed unsupervised GSL paradigm.

**Figure 1: Concept maps of (a) the existing supervised GSL paradigm and (b) our proposed unsupervised GSL paradigm.**

# Method

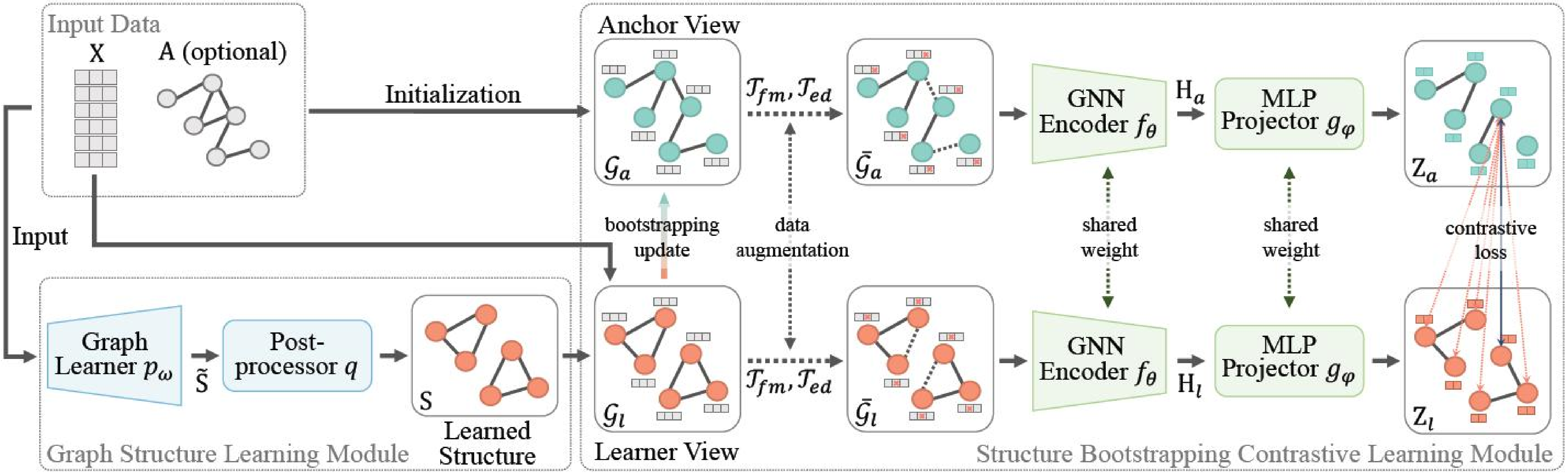


**Figure 2: The overall pipeline of SUBLIME. In the graph structure learning module, the graph learner $p_\omega$ generates the sketched adjacency matrix $\tilde{\mathrm{S}}$, and then the post processor $q$ converts $\tilde{\mathrm{S}}$ into the learned structure S. After that, the structure bootstrapping contrastive learning module optimizes S by maximizing the agreement between the learner view and anchor view.**

# Method

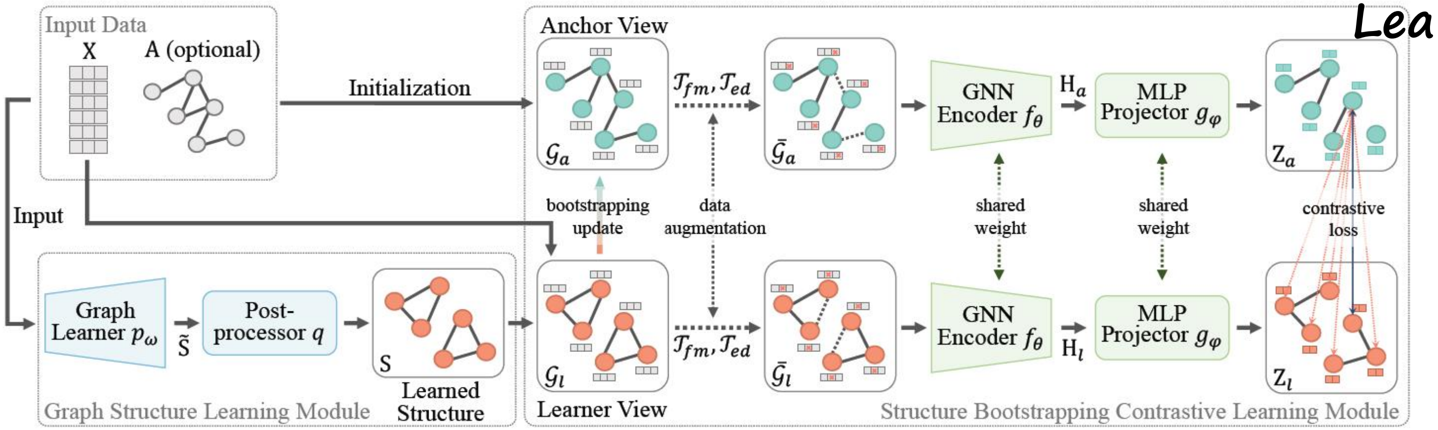


Figure 2: The overall pipeline of SUBLIME. In the graph structure learning module, the graph learner $p_\omega$ generates the sketched adjacency matrix $\tilde{S}$, and then the post processor $q$ converts $\tilde{S}$ into the learned structure S. After that, the structure bootstrapping contrastive learning module optimizes S by maximizing the agreement between the learner view and anchor view.

Graph Learner

Full Graph Parameterization, FGP

$$\tilde{S} = p_\omega^{FGP} = \sigma(\Omega), \qquad (1)$$

where $\omega = \Omega \in \mathbb{R}^{n \times n}$ is a parameter matrix

$$\tilde{S} = p_\omega^{ML}(X, A) = \phi(h_\omega(X, A)) = \phi(E), \qquad (2)$$

Attentive Learner

$$E^{(l)} = h_w^{(l)}(E^{(l-1)}) = \sigma([e_1^{(l-1)} \odot \omega^{(l)}, \cdots, e_n^{(l-1)} \odot \omega^{(l)}]^\top), \quad (3)$$

MLP Learner

$$E^{(l)} = h_w^{(l)}(E^{(l-1)}) = \sigma(E^{(l-1)}\Omega^{(l)}), \qquad (4)$$

GNN

$$E^{(l)} = h_w^{(l)}(E^{(l-1)}, A) = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}E^{(l-1)}\Omega^{(l)}\right), \qquad (5)$$

# Method

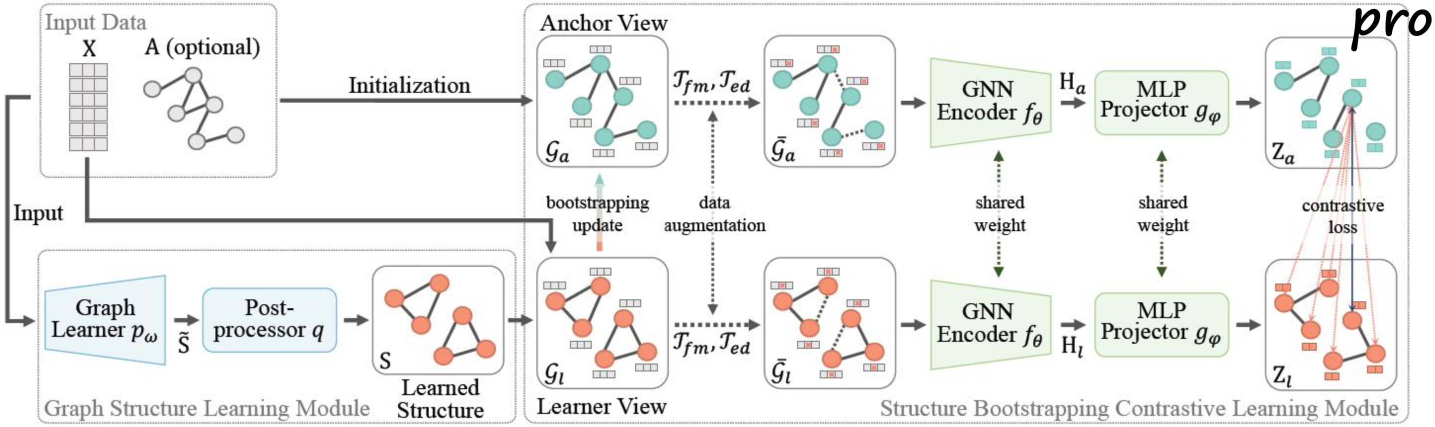


Figure 2: The overall pipeline of SUBLIME. In the graph structure learning module, the graph learner $p_\omega$ generates the sketched adjacency matrix $\tilde{S}$, and then the post processor $q$ converts $\tilde{S}$ into the learned structure S. After that, the structure bootstrapping contrastive learning module optimizes S by maximizing the agreement between the learner view and anchor view.

## Post-processor

### Sparsificatio

$$\tilde{S}_{ij}^{(sp)} = q_{sp}\left(\tilde{S}_{ij}\right) = \begin{cases} \tilde{S}_{ij}, & \tilde{S}_{ij} \in \text{top-k}(\tilde{S}_i), \\ 0, & \tilde{S}_{ij} \notin \text{top-k}(\tilde{S}_i), \end{cases} \qquad (6)$$

### Symmetrization and

'

$$\tilde{S}^{(sym)} = q_{sym}\left(q_{act}\left(\tilde{S}^{(sp)}\right)\right) = \frac{\sigma_q\left(\tilde{S}^{(sp)}\right) + \sigma_q\left(\tilde{S}^{(sp)}\right)^{\top}}{2}, \qquad (7)$$

define $\sigma_q(\cdot)$ as ReLU function

### Normalizati

$$S = q_{norm}\left(\tilde{S}^{(sym)}\right) = \left(\tilde{D}^{(sym)}\right)^{-\frac{1}{2}} \tilde{S}^{(sym)} \left(\tilde{D}^{(sym)}\right)^{-\frac{1}{2}}, \qquad (8)$$

# Method

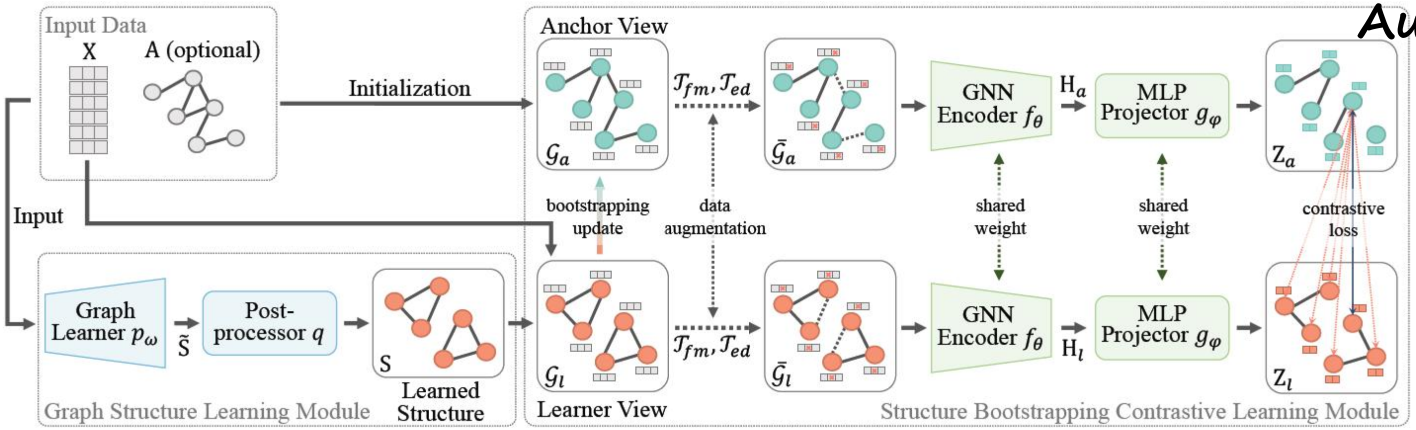


Figure 2: The overall pipeline of SUBLIME. In the graph structure learning module, the graph learner $p_\omega$ generates the sketched adjacency matrix $\tilde{S}$, and then the post processor $q$ converts $\tilde{S}$ into the learned structure S. After that, the structure bootstrapping contrastive learning module optimizes S by maximizing the agreement between the learner view and anchor view.

Data Augmentation

Feature masking

$$\overline{X} = \mathcal{T}_{fm}(X) = [x_1 \odot m^{(x)}, \cdots, x_n \odot m^{(x)}]^\top, \tag{9}$$

Edge dropping

$$\overline{A} = \mathcal{T}_{ed}(A) = A \odot M^{(a)}, \tag{10}$$

$$\overline{\mathcal{G}}_l = (\mathcal{T}_{ed}(S), \mathcal{T}_{fm}(X)), \ \overline{\mathcal{G}}_a = (\mathcal{T}_{ed}(A_a), \mathcal{T}_{fm}(X)), \tag{11}$$

# Method

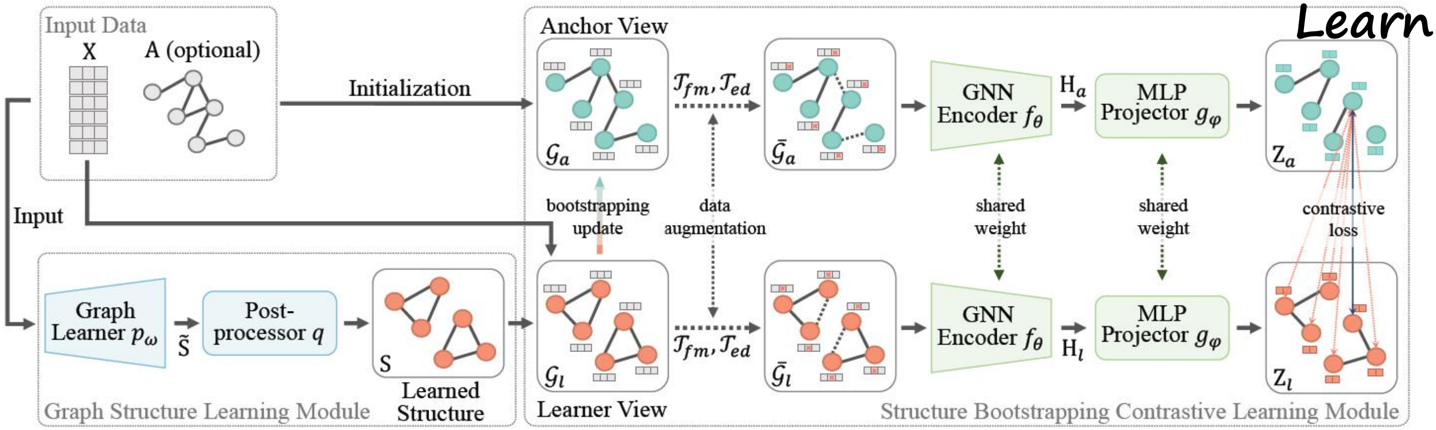


Figure 2: The overall pipeline of SUBLIME. In the graph structure learning module, the graph learner $p_\omega$ generates the sketched adjacency matrix $\tilde{S}$, and then the post processor $q$ converts $\tilde{S}$ into the learned structure S. After that, the structure bootstrapping contrastive learning module optimizes S by maximizing the agreement between the learner view and anchor view.

## Node-level Contrastive Learning

GNN-based en

$$H_l = f_\theta(\overline{\mathcal{G}}_l),\ H_a = f_\theta(\overline{\mathcal{G}}_a), \tag{12}$$

MLP-based pr

$$Z_l = g_\varphi(H_l),\ Z_a = g_\varphi(H_a), \tag{13}$$

Node-level contrastive loss f

$$\mathcal{L} = \frac{1}{2n}\sum_{i=1}^{n}\left[\ell(z_{l,i}, z_{a,i}) + \ell(z_{a,i}, z_{l,i})\right],$$
$$\ell(z_{l,i}, z_{a,i}) = \log\frac{e^{\mathrm{sim}(z_{l,i}, z_{a,i})/t}}{\sum_{k=1}^{n} e^{\mathrm{sim}(z_{l,i}, z_{a,k})/t}}, \tag{14}$$

$\mathrm{sim}(\cdot,\cdot)$ is the cosine similarity function

## Structure Bootstrapping Mechanism

$$A_a \leftarrow \tau A_a + (1-\tau)S. \tag{15}$$

# Experiments

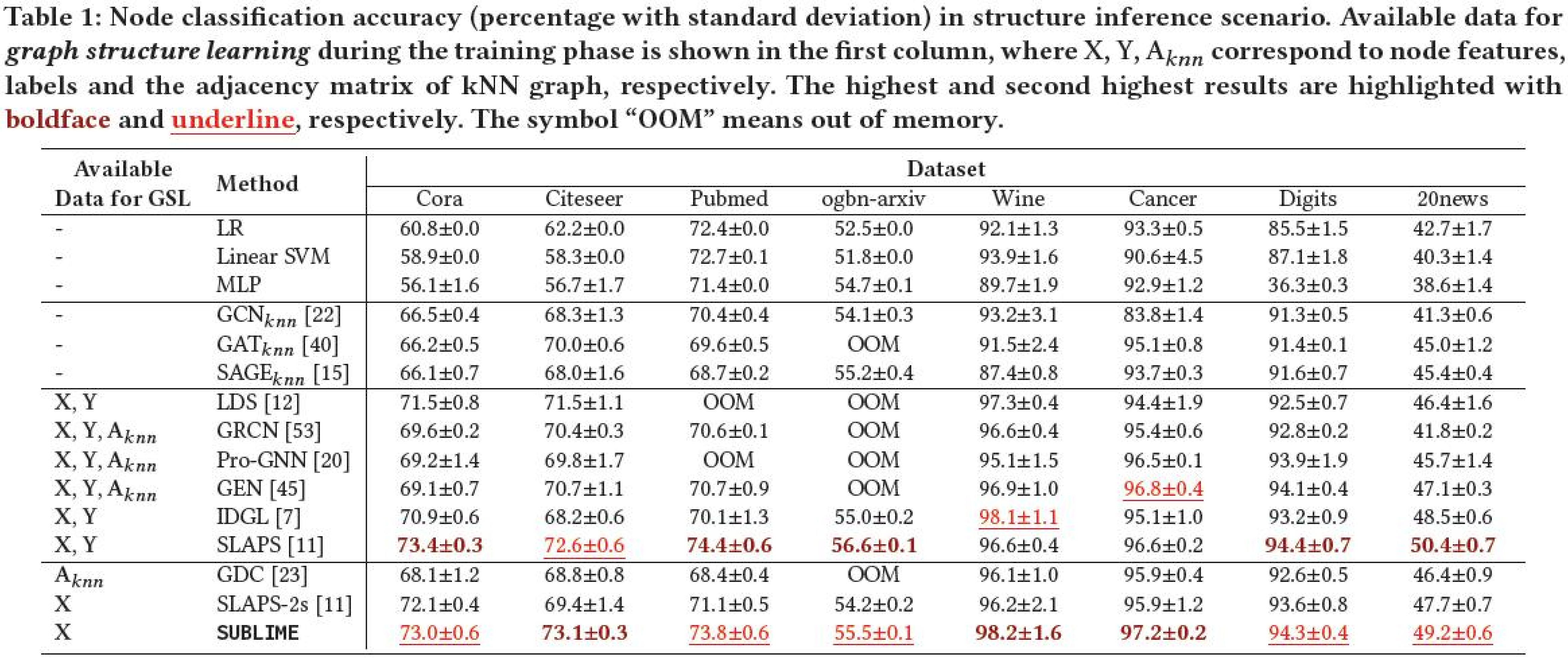


Table 1: Node classification accuracy (percentage with standard deviation) in structure inference scenario. Available data for *graph structure learning* during the training phase is shown in the first column, where X, Y, $A_{knn}$ correspond to node features, labels and the adjacency matrix of kNN graph, respectively. The highest and second highest results are highlighted with **boldface** and underline, respectively. The symbol "OOM" means out of memory.

| Available Data for GSL | Method | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Cora | Citeseer | Pubmed | ogbn-arxiv | Wine | Cancer | Digits | 20news |
| - | LR | 60.8±0.0 | 62.2±0.0 | 72.4±0.0 | 52.5±0.0 | 92.1±1.3 | 93.3±0.5 | 85.5±1.5 | 42.7±1.7 |
| - | Linear SVM | 58.9±0.0 | 58.3±0.0 | 72.7±0.1 | 51.8±0.0 | 93.9±1.6 | 90.6±4.5 | 87.1±1.8 | 40.3±1.4 |
| - | MLP | 56.1±1.6 | 56.7±1.7 | 71.4±0.0 | 54.7±0.1 | 89.7±1.9 | 92.9±1.2 | 36.3±0.3 | 38.6±1.4 |
| - | $GCN_{knn}$ [22] | 66.5±0.4 | 68.3±1.3 | 70.4±0.4 | 54.1±0.3 | 93.2±3.1 | 83.8±1.4 | 91.3±0.5 | 41.3±0.6 |
| - | $GAT_{knn}$ [40] | 66.2±0.5 | 70.0±0.6 | 69.6±0.5 | OOM | 91.5±2.4 | 95.1±0.8 | 91.4±0.1 | 45.0±1.2 |
| - | $SAGE_{knn}$ [15] | 66.1±0.7 | 68.0±1.6 | 68.7±0.2 | 55.2±0.4 | 87.4±0.8 | 93.7±0.3 | 91.6±0.7 | 45.4±0.4 |
| X, Y | LDS [12] | 71.5±0.8 | 71.5±1.1 | OOM | OOM | 97.3±0.4 | 94.4±1.9 | 92.5±0.7 | 46.4±1.6 |
| X, Y, $A_{knn}$ | GRCN [53] | 69.6±0.2 | 70.4±0.3 | 70.6±0.1 | OOM | 96.6±0.4 | 95.4±0.6 | 92.8±0.2 | 41.8±0.2 |
| X, Y, $A_{knn}$ | Pro-GNN [20] | 69.2±1.4 | 69.8±1.7 | OOM | OOM | 95.1±1.5 | 96.5±0.1 | 93.9±1.9 | 45.7±1.4 |
| X, Y, $A_{knn}$ | GEN [45] | 69.1±0.7 | 70.7±1.1 | 70.7±0.9 | OOM | 96.9±1.0 | 96.8±0.4 | 94.1±0.4 | 47.1±0.3 |
| X, Y | IDGL [7] | 70.9±0.6 | 68.2±0.6 | 70.1±1.3 | 55.0±0.2 | 98.1±1.1 | 95.1±1.0 | 93.2±0.9 | 48.5±0.6 |
| X, Y | SLAPS [11] | **73.4±0.3** | 72.6±0.6 | **74.4±0.6** | **56.6±0.1** | 96.6±0.4 | 96.6±0.2 | **94.4±0.7** | **50.4±0.7** |
| $A_{knn}$ | GDC [23] | 68.1±1.2 | 68.8±0.8 | 68.4±0.4 | OOM | 96.1±1.0 | 95.9±0.4 | 92.6±0.5 | 46.4±0.9 |
| X | SLAPS-2s [11] | 72.1±0.4 | 69.4±1.4 | 71.1±0.5 | 54.2±0.2 | 96.2±2.1 | 95.9±1.2 | 93.6±0.8 | 47.7±0.7 |
| X | SUBLIME | 73.0±0.6 | **73.1±0.3** | 73.8±0.6 | 55.5±0.1 | **98.2±1.6** | **97.2±0.2** | 94.3±0.4 | 49.2±0.6 |

# Experiments

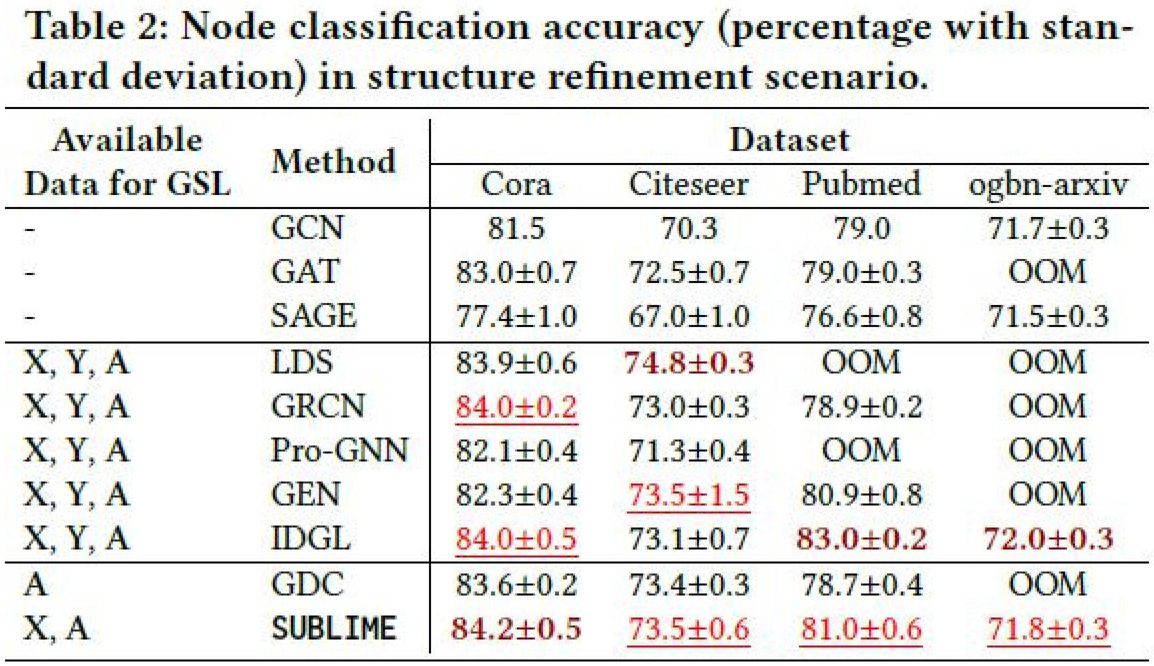


Table 2: Node classification accuracy (percentage with standard deviation) in structure refinement scenario.

| Available Data for GSL | Method | Cora | Citeseer | Pubmed | ogbn-arxiv |
|---|---|---|---|---|---|
| | | Dataset | | | |
| - | GCN | 81.5 | 70.3 | 79.0 | 71.7±0.3 |
| - | GAT | 83.0±0.7 | 72.5±0.7 | 79.0±0.3 | OOM |
| - | SAGE | 77.4±1.0 | 67.0±1.0 | 76.6±0.8 | 71.5±0.3 |
| X, Y, A | LDS | 83.9±0.6 | **74.8±0.3** | OOM | OOM |
| X, Y, A | GRCN | 84.0±0.2 | 73.0±0.3 | 78.9±0.2 | OOM |
| X, Y, A | Pro-GNN | 82.1±0.4 | 71.3±0.4 | OOM | OOM |
| X, Y, A | GEN | 82.3±0.4 | 73.5±1.5 | 80.9±0.8 | OOM |
| X, Y, A | IDGL | 84.0±0.5 | 73.1±0.7 | **83.0±0.2** | **72.0±0.3** |
| A | GDC | 83.6±0.2 | 73.4±0.3 | 78.7±0.4 | OOM |
| X, A | SUBLIME | **84.2±0.5** | 73.5±0.6 | 81.0±0.6 | 71.8±0.3 |

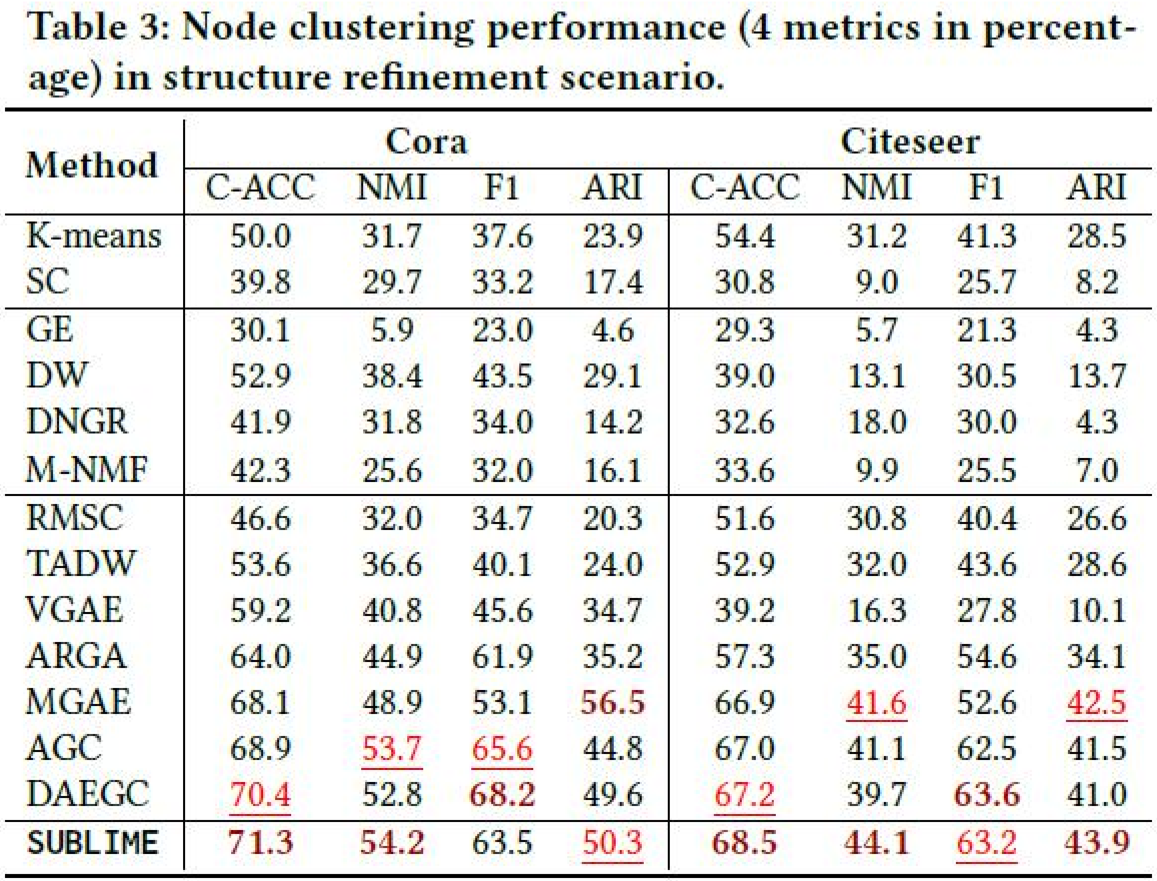


Table 3: Node clustering performance (4 metrics in percentage) in structure refinement scenario.

| Method | Cora | | | | Citeseer | | | |
|---|---|---|---|---|---|---|---|---|
| | C-ACC | NMI | F1 | ARI | C-ACC | NMI | F1 | ARI |
| K-means | 50.0 | 31.7 | 37.6 | 23.9 | 54.4 | 31.2 | 41.3 | 28.5 |
| SC | 39.8 | 29.7 | 33.2 | 17.4 | 30.8 | 9.0 | 25.7 | 8.2 |
| GE | 30.1 | 5.9 | 23.0 | 4.6 | 29.3 | 5.7 | 21.3 | 4.3 |
| DW | 52.9 | 38.4 | 43.5 | 29.1 | 39.0 | 13.1 | 30.5 | 13.7 |
| DNGR | 41.9 | 31.8 | 34.0 | 14.2 | 32.6 | 18.0 | 30.0 | 4.3 |
| M-NMF | 42.3 | 25.6 | 32.0 | 16.1 | 33.6 | 9.9 | 25.5 | 7.0 |
| RMSC | 46.6 | 32.0 | 34.7 | 20.3 | 51.6 | 30.8 | 40.4 | 26.6 |
| TADW | 53.6 | 36.6 | 40.1 | 24.0 | 52.9 | 32.0 | 43.6 | 28.6 |
| VGAE | 59.2 | 40.8 | 45.6 | 34.7 | 39.2 | 16.3 | 27.8 | 10.1 |
| ARGA | 64.0 | 44.9 | 61.9 | 35.2 | 57.3 | 35.0 | 54.6 | 34.1 |
| MGAE | 68.1 | 48.9 | 53.1 | **56.5** | 66.9 | 41.6 | 52.6 | 42.5 |
| AGC | 68.9 | 53.7 | 65.6 | 44.8 | 67.0 | 41.1 | 62.5 | 41.5 |
| DAEGC | 70.4 | 52.8 | **68.2** | 49.6 | 67.2 | 39.7 | **63.6** | 41.0 |
| SUBLIME | **71.3** | **54.2** | 63.5 | 50.3 | **68.5** | **44.1** | 63.2 | **43.9** |

# Experiments

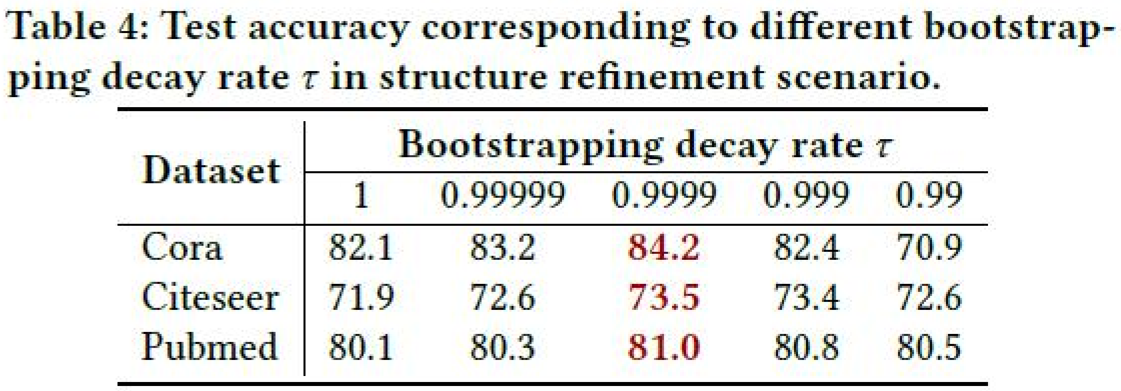
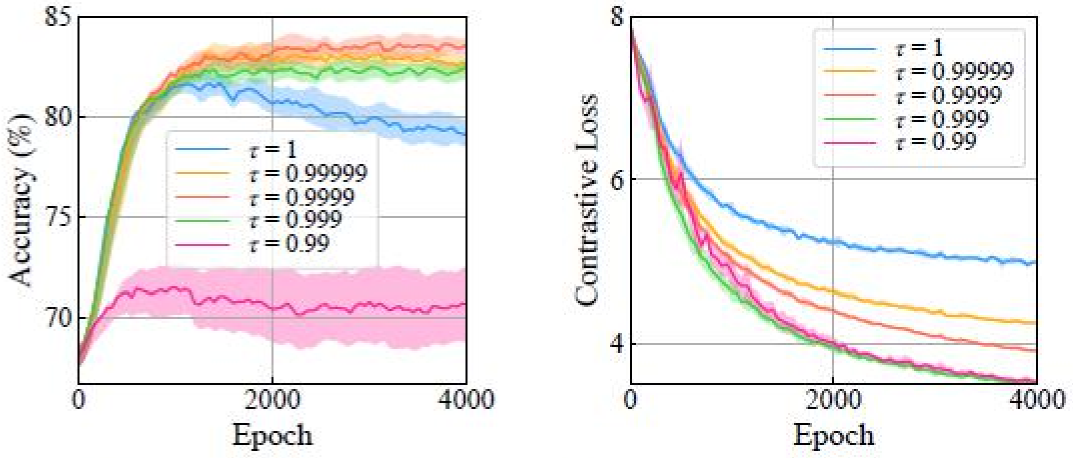


Table 4: Test accuracy corresponding to different bootstrapping decay rate $\tau$ in structure refinement scenario.

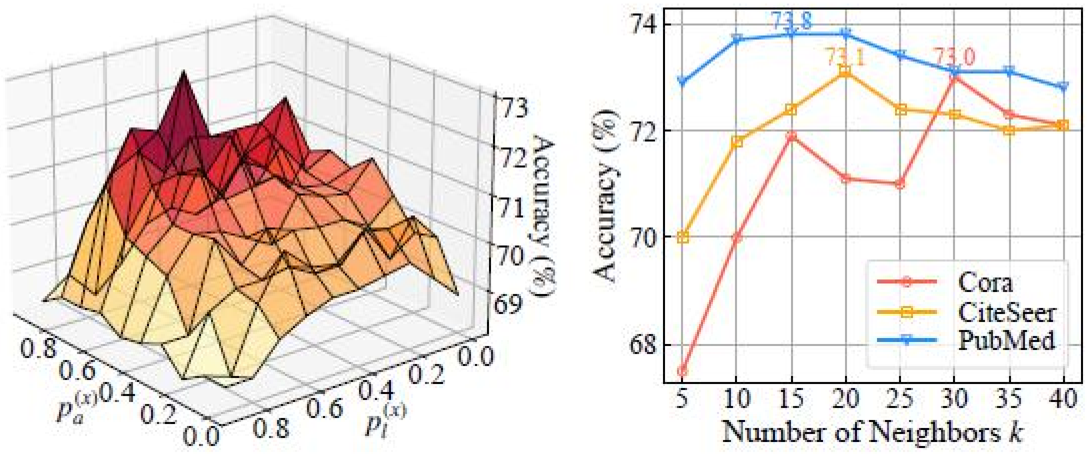
| Dataset | Bootstrapping decay rate $\tau$ | | | | |
|---|---|---|---|---|---|
| | 1 | 0.99999 | 0.9999 | 0.999 | 0.99 |
| Cora | 82.1 | 83.2 | 84.2 | 82.4 | 70.9 |
| Citeseer | 71.9 | 72.6 | 73.5 | 73.4 | 72.6 |
| Pubmed | 80.1 | 80.3 | 81.0 | 80.8 | 80.5 |



(a) Test accuracy w.r.t. epoch. (b) Contrastive loss value w.r.t. epoch.

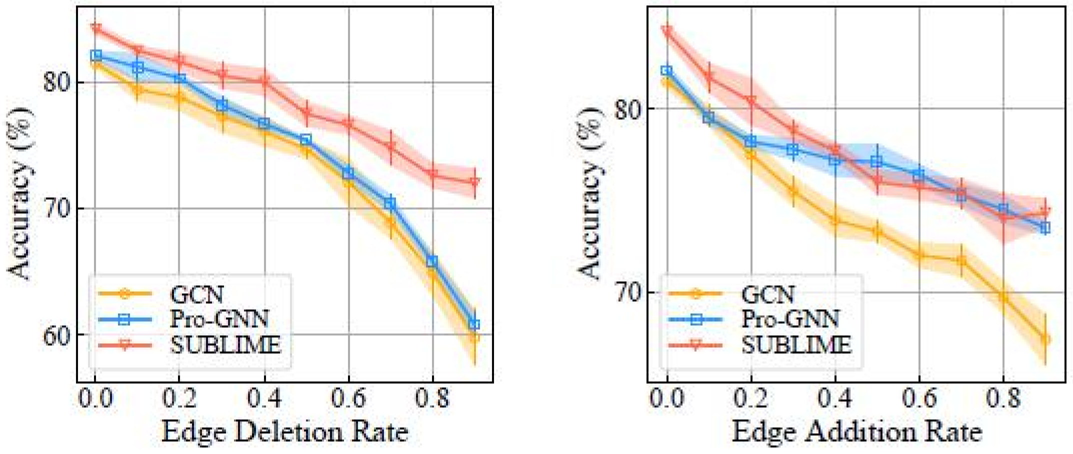
Figure 3: Curves of training process on Cora dataset.

# Experiments



(a) Accuracy w.r.t. feature masking rates. (b) Accuracy w.r.t. number of neighbors.

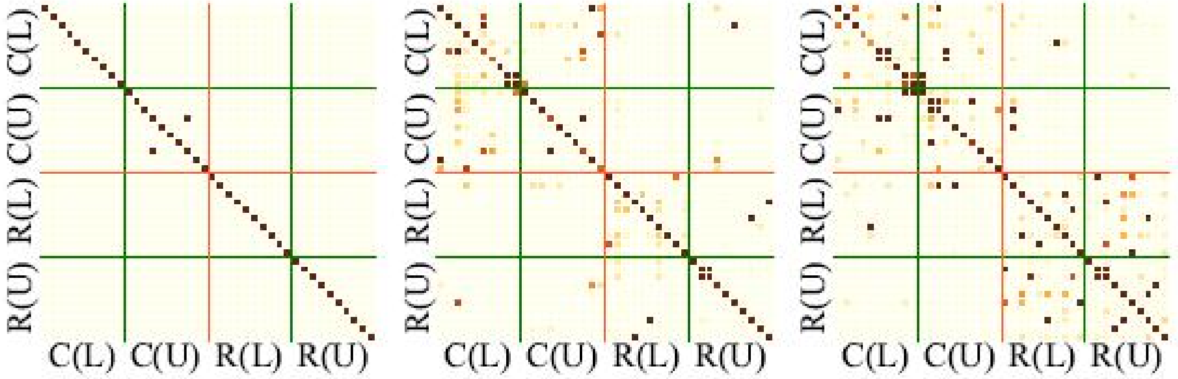
Figure 4: Sensitivity of hyper-parameters $p^{(x)}$ and $k$.



(a) Accuracy w.r.t. edge deletion rate. (b) Accuracy w.r.t. edge addition rate.

Figure 5: Test accuracy in the scenarios where graph structures are perturbed by edge deletion or addition.

# Experiments



(a) Original graph. (b) Graph learned by Pro-GNN. (c) Graph learned by SUBLIME.

Figure 6: Heatmaps of the subgraph adjacency matrices of (a) the original graph with self-loop, the graph learned by (b) Pro-GNN and (c) SUBLIME on Cora dataset. A block in darker color indicates a larger edge weight between two nodes.

# Thank you!