Simple Unsupervised Graph Representation Learning

Yujie Mo*, Liang Peng*, Jie Xu, Xiaoshuang Shi†, Xiaofeng Zhu

School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
moyujie2017@gmail.com, larrypengliang@gmail.com, jiexuwork@outlook.com, xsshi2013@gmail.com, seanzhuxf@gmail.com

Code : https://github.com/YujieMo/SUGRL

AAAI_2022

Reported by Xinsheng Wang

Artificial

1.Introduction

2.Method

3.Experiments

Leibniz-Institut für Sozialwissenschaften

Introduction

对比学习定义：尽可能的缩小相似样本的距离，拉大正负样本的距离。

对比学习一般泛式 一般包含三种样本：anchor、正样本、负样本

对任意数据 x ，对比学习的目标是学习一个编码器 f 使得：

$$score(f(x), f(x^+)) >> score(f(x), f(x^-))$$

其中 x^+ 是和 x 相似的正样本， x^- 是和 x 不相似的负样本，score是一个度量函数来衡量样本间的相似度。

如果用向量内积来计算两个样本的相似度，则对比学习的损失函数可以表示成：

$$L_N = -\mathbb{E}_X\left[\log \frac{\exp\left(f(x)^T f(x^+)\right)}{\exp(f(x)^T f(x^+)) + \sum_{j=1}^{N-1} \exp\left(f(x)^T f(x_j^-)\right)}\right]$$

其中对应样本 x 有1个样本和N-1个负样本。可以发现，这个形式类似于交叉熵损失函数，学习的目标就是让 x 的特征和正样本的特征更相似，同时和N-1个负样本的特征更不相似。在对比学习的相关文献中把这一损失函数称作InfoNCE损失。也有一些其他的工作把这一损失函数称为multi-class n-pair loss或者ranking-based NCE。

Method

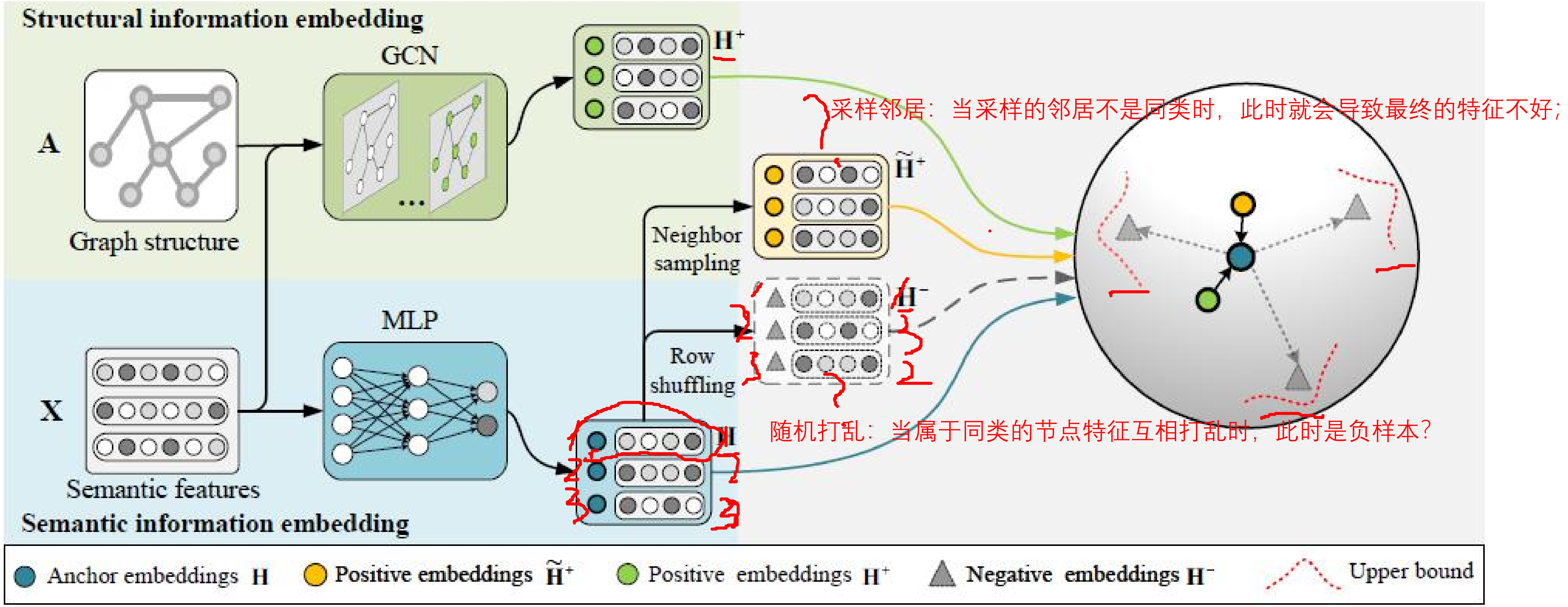

Figure 2: The flowchart of the proposed SUGRL method. Specifically, given the semantic features X and its graph structure A, the SUGRL employs a MLP network on X with the semantic information to generate the anchor embeddings H, and employs GCN to generate positive embeddings H^+ with the structural information as well as the neighbor sampling method to generate positive embeddings $\tilde{H}^+$ with the neighbor information. The SUGRL also employs the row-wise random permutation method on H to generate negative embeddings H^-, and further designs a multiplet loss to achieve that the anchor embeddings are close to positive embeddings and far away from negative embeddings.

Method

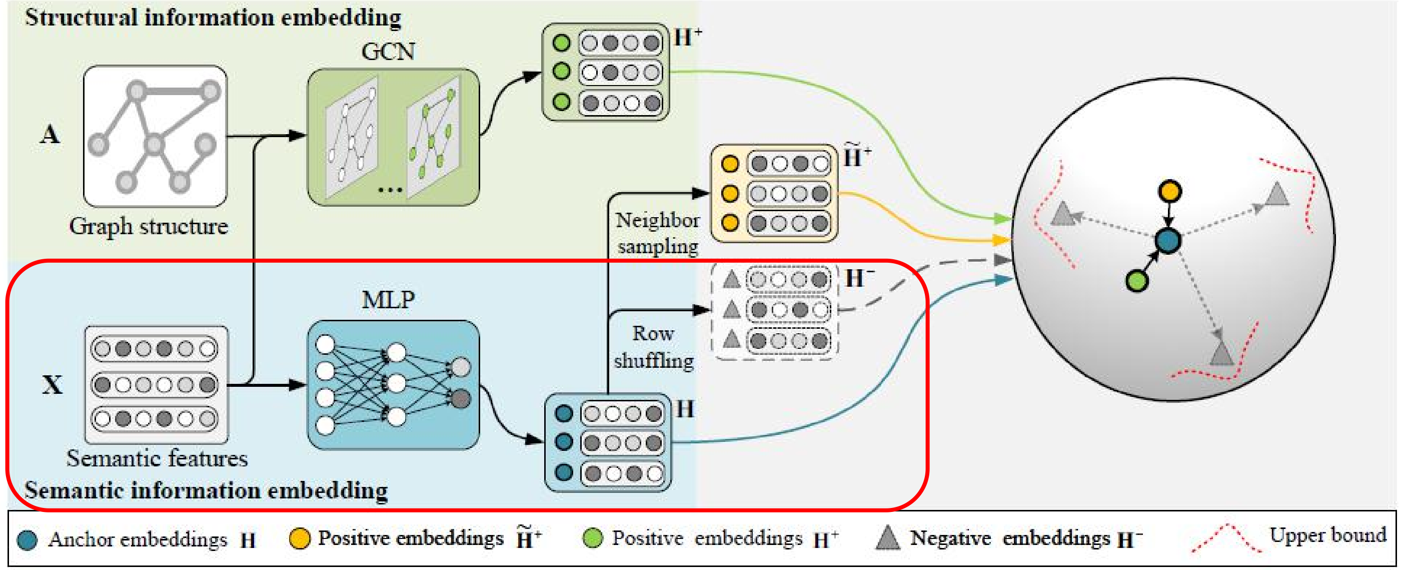

Figure 2: The flowchart of the proposed SUGRL method. Specifically, given the semantic features X and its graph structure A, the SUGRL employs a MLP network on X with the semantic information to generate the anchor embeddings H, and employs GCN to generate positive embeddings $\mathbf{H}^{+}$ with the structural information as well as the neighbor sampling method to generate positive embeddings $\widetilde{\mathbf{H}}^{+}$ with the neighbor information. The SUGRL also employs the row-wise random permutation method on H to generate negative embeddings $\mathbf{H}^{-}$, and further designs a multiplet loss to achieve that the anchor embeddings are close to positive embeddings and far away from negative embeddings.

Anchor and negative embedding generation

$$\mathbf{X}^{(l+1)} = Dropout\left(\sigma\left(\mathbf{X}^{(l)}\mathbf{W}^{(l)}\right)\right), \tag{1}$$

$$\mathbf{H} = \mathbf{X}^{(l+1)}\mathbf{W}^{(l+1)}, \tag{2}$$

where $\mathbf{X}^{(0)} = \mathbf{X}$, σ is an activation function, and $\mathbf{W}^{(l)}$ is the weight of the l^{th} layer.

$$\mathbf{H}^{-} = Shuffle\left([\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]\right). \tag{3}$$

By contrast, we directly row-shuffle anchor embeddings to obtain negative embeddings

Method

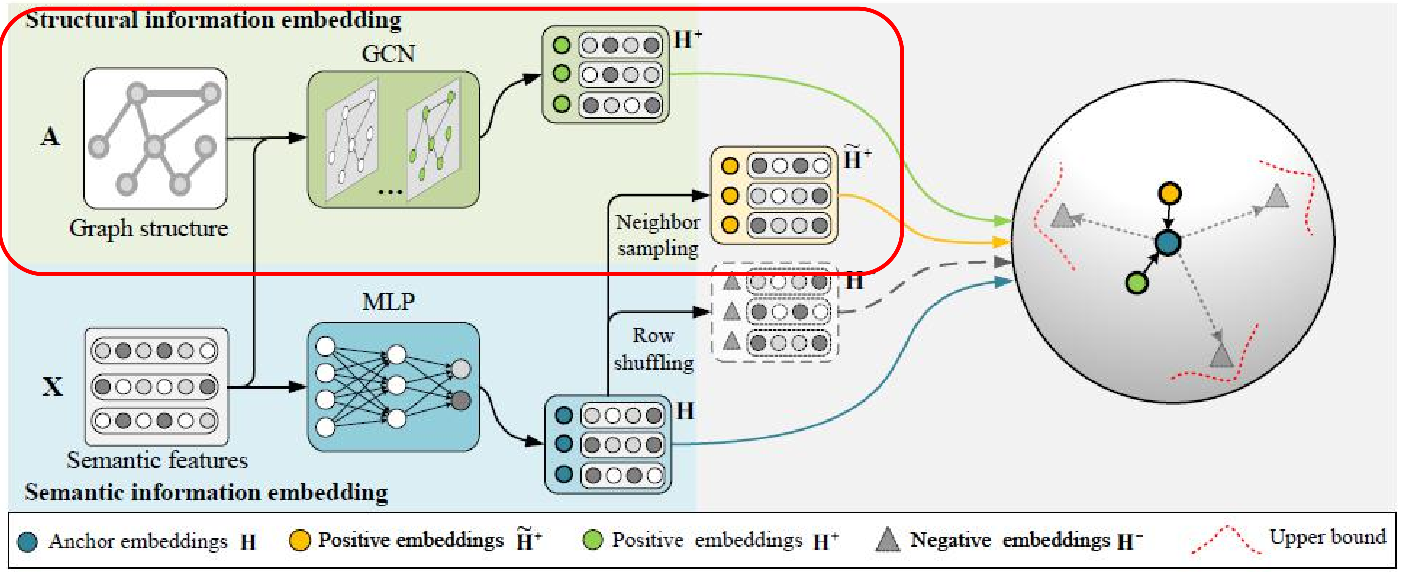

Figure 2: The flowchart of the proposed SUGRL method. Specifically, given the semantic features X and its graph structure A, the SUGRL employs a MLP network on X with the semantic information to generate the anchor embeddings H, and employs GCN to generate positive embeddings H^+ with the structural information as well as the neighbor sampling method to generate positive embeddings $\widetilde{H}^+$ with the neighbor information. The SUGRL also employs the row-wise random permutation method on H to generate negative embeddings H^-, and further designs a multiplet loss to achieve that the anchor embeddings are close to positive embeddings and far away from negative embeddings.

Positive embedding generation

Structural information

$$\mathrm{H}^{+(l+1)} = \sigma\left(\widehat{\mathrm{A}}\mathrm{H}^{+(l)}\mathrm{W}^{(l)}\right), \tag{4}$$

where $\mathrm{H}^{+(0)} = \mathrm{X}$ and $\mathrm{H}^{+(l)}$ means the l^{th} layer features. $\widehat{\mathrm{A}} = \hat{\mathrm{D}}^{-1/2}\tilde{\mathrm{A}}\hat{\mathrm{D}}^{-1/2} \in \mathbb{R}^{N\times N}$ is a symmetrically normalized adjacency matrix, $\hat{\mathrm{D}} \in \mathbb{R}^{N\times N}$ is the degree matrix of $\tilde{\mathrm{A}} = \mathrm{A} + \mathrm{I}_N$, I_N is the identity matrix. It is

Neighbor information

$$\tilde{\mathrm{h}}_i^+ = \frac{1}{m}\sum_{j=1}^{m}\{\mathrm{h}_j \mid v_j \in \mathcal{N}_i\}, \tag{5}$$

where m is the number of sampled neighbors, $\mathcal{N}_i$ represents 1-hop neighborhood set of node v_i.

Method

Multiplet loss

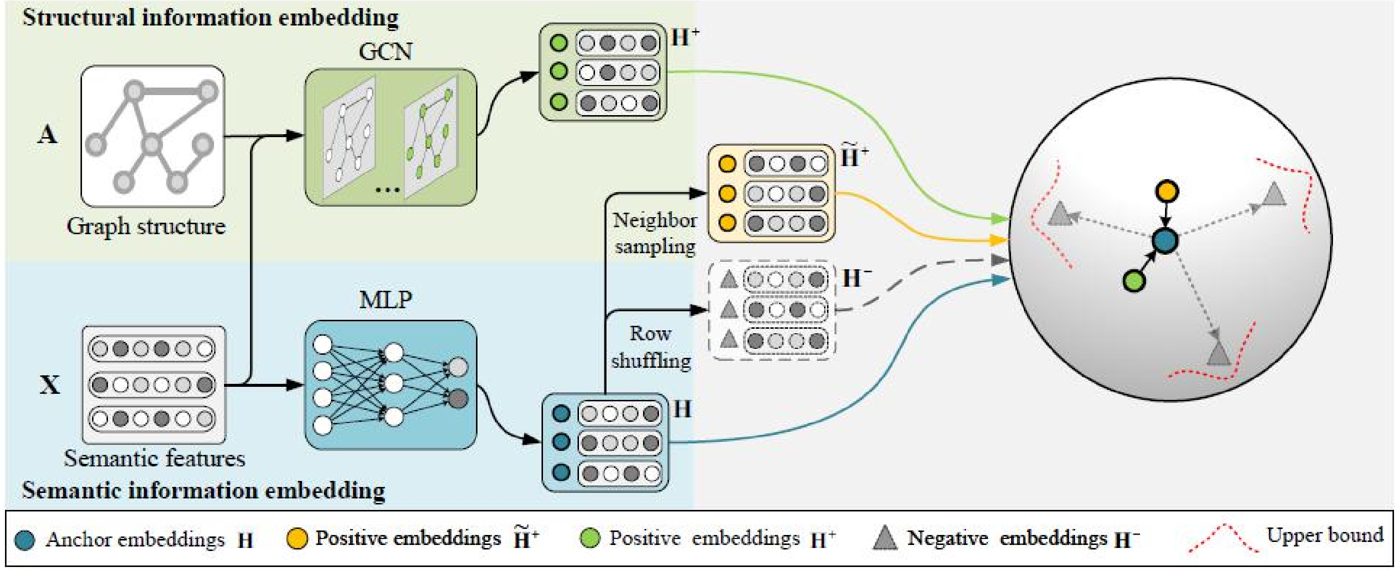

Figure 2: The flowchart of the proposed SUGRL method. Specifically, given the semantic features X and its graph structure A, the SUGRL employs a MLP network on X with the semantic information to generate the anchor embeddings H, and employs GCN to generate positive embeddings $\mathbf{H}^{+}$ with the structural information as well as the neighbor sampling method to generate positive embeddings $\widetilde{\mathbf{H}}^{+}$ with the neighbor information. The SUGRL also employs the row-wise random permutation method on H to generate negative embeddings $\mathbf{H}^{-}$, and further designs a multiplet loss to achieve that the anchor embeddings are close to positive embeddings and far away from negative embeddings.

$$\mathcal{L}_S = \frac{1}{k}\sum_{i=1}^{k}\left\{d\left(\mathrm{h}, \mathrm{h}^{+}\right)^2 - d\left(\mathrm{h}, \mathrm{h}_i^{-}\right)^2 + \alpha\right\}_{+}, \quad (8)$$

$$\mathcal{L}_N = \frac{1}{k}\sum_{j=1}^{k}\left\{d\left(\mathrm{h}, \widetilde{\mathrm{h}}^{+}\right)^2 - d\left(\mathrm{h}, \mathrm{h}_j^{-}\right)^2 + \alpha\right\}_{+}. \quad (9)$$

where $d(\cdot)$ is a similarity measurement

α is a non-negative value

where $\{\cdot\}_{+} = \max\{\cdot, 0\}$, and k is the number of negative samples.

$$\mathcal{L}_U = -\frac{1}{k}\sum_{i=1}^{k}\left\{d\left(\mathrm{h}, \mathrm{h}^{+}\right)^2 - d\left(\mathrm{h}, \mathrm{h}_i^{-}\right)^2 + \alpha + \beta\right\}_{-}, \quad (11)$$

where $\{\cdot\}_{-} = \min\{\cdot, 0\}$

$$\mathcal{L} = \omega_1\mathcal{L}_S + \omega_2\mathcal{L}_N + \mathcal{L}_U, \quad (12)$$

Experiments

Method	Cora		CiteSeer		PubMed		Photo	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Raw Feature	47.9 ± 0.4	–	49.3 ± 0.3	–	69.1 ± 0.2	–	78.5 ± 0.2	–
Deep Walk	67.2 ± 0.2	19.2	43.2 ± 0.4	4.6	65.3 ± 0.5	144.2	89.4 ± 0.1	76.8
GCN	81.5 ± 0.2	3.1	70.3 ± 0.4	1.4	79.0 ± 0.5	6.1	91.6 ± 0.3	6.7
GAT	83.0 ± 0.2	16.9	72.5 ± 0.3	4.2	79.0 ± 0.5	62.5	91.8 ± 0.1	50.0
GAE	74.9 ± 0.4	24.5	65.6 ± 0.5	8.1	74.2 ± 0.3	165.4	91.0 ± 0.1	108.4
VGAE	76.3 ± 0.2	26.9	66.8 ± 0.2	8.7	75.8 ± 0.4	166.3	91.5 ± 0.2	107.8
DGI	82.3 ± 0.5	10.5	71.5 ± 0.4	3.1	79.4 ± 0.3	128.1	91.3 ± 0.1	54.1
GMI	83.0 ± 0.2	100.1	72.4 ± 0.2	24.3	79.9 ± 0.4	1104.2	90.6 ± 0.2	461.3
GRACE	83.1 ± 0.2	6.8	72.1 ± 0.1	2.5	79.6 ± 0.5	196.9	91.9 ± 0.3	53.4
MVGRL	82.9 ± 0.3	67.1	72.6 ± 0.4	18.3	80.1 ± 0.7	669.2	91.7 ± 0.1	272.3
GCA	81.8 ± 0.2	11.1	71.9 ± 0.4	4.2	81.0 ± 0.3	312.1	92.4 ± 0.4	65.1
GIC	81.7 ± 0.8	8.6	71.9 ± 0.9	3.6	77.4 ± 0.5	15.1	91.6 ± 0.1	15.2
SUGRL	$\mathbf{83.4 \pm 0.5}$	3.8	$\mathbf{73.0 \pm 0.4}$	0.9	$\mathbf{81.9 \pm 0.3}$	9.5	$\mathbf{93.2 \pm 0.4}$	5.6

Table 1: Classification accuracy (%) and execution time (seconds) of all methods on four datasets.

Experiments

Method	Computers		Ogbn-arxiv		Ogbn-mag		Ogbn-products	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
Raw Feature	73.8 ± 0.1	–	56.3 ± 0.3	–	22.1 ± 0.3	–	59.7 ± 0.2	–
Deep Walk	85.3 ± 0.1	2.2	63.6 ± 0.4	5.1	25.6 ± 0.3	12.1	73.2 ± 0.2	30.5
GCN	84.5 ± 0.3	0.2	70.4 ± 0.3	0.1	30.1 ± 0.3	0.7	81.6 ± 0.4	2.6
GAT	85.7 ± 0.1	1.5	$\mathbf{70.6 \pm 0.3}$	2.5	30.5 ± 0.3	6.1	82.4 ± 0.4	14.6
GAE	85.1 ± 0.4	4.1	63.6 ± 0.5	9.4	27.1 ± 0.3	22.5	72.1 ± 0.1	56.8
VGAE	85.8 ± 0.3	4.0	64.8 ± 0.2	9.5	27.9 ± 0.2	22.7	72.9 ± 0.2	57.3
DGI	87.8 ± 0.2	2.0	65.1 ± 0.4	4.5	31.4 ± 0.3	10.6	77.9 ± 0.2	26.8
GMI	82.2 ± 0.4	15.3	68.2 ± 0.2	42.0	29.5 ± 0.1	123.1	76.8 ± 0.3	293.4
GRACE	86.8 ± 0.2	2.2	68.7 ± 0.4	7.2	31.5 ± 0.3	19.2	77.4 ± 0.4	43.3
MVGRL	86.9 ± 0.1	10.4	68.1 ± 0.1	24.6	31.6 ± 0.4	67.3	78.1 ± 0.1	213.7
GCA	87.7 ± 0.1	2.6	68.2 ± 0.2	7.1	31.4 ± 0.3	37.5	78.4 ± 0.3	81.2
GIC	84.9 ± 0.2	0.4	68.4 ± 0.4	1.1	31.7 ± 0.2	5.1	75.8 ± 0.2	8.8
SUGRL	$\mathbf{88.9 \pm 0.2}$	0.2	68.8 ± 0.4	0.1	31.9 ± 0.3	0.4	$\mathbf{82.6 \pm 0.4}$	2.1
SUGRL-batch	–	–	69.3 ± 0.2	0.2	$\mathbf{32.4 \pm 0.1}$	0.4	81.2 ± 0.1	2.2

Table 2: Classification accuracy (%) and execution time (minutes) of all methods on four datasets.

Experiments

$\mathcal{L}_S$	$\mathcal{L}_N$	$\mathcal{L}_U$	Cora	CiteSeer	PubMed	Photo	Computers	Ogbn-arxiv	Ogbn-mag	Ogbn-products
√	–	–	73.8 ± 0.6	71.7 ± 0.5	70.7 ± 0.3	91.0 ± 0.2	84.4 ± 0.2	68.1 ± 0.1	31.2 ± 0.2	82.3 ± 0.1
–	√	–	78.1 ± 0.4	71.8 ± 0.3	80.5 ± 0.3	79.7 ± 0.2	72.5 ± 0.4	67.9 ± 0.2	31.1 ± 0.1	82.1 ± 0.1
√	√	–	78.5 ± 0.4	71.9 ± 0.4	81.6 ± 0.3	91.6 ± 0.3	86.6 ± 0.4	68.5 ± 0.1	31.6 ± 0.1	82.4 ± 0.1
√	–	√	81.5 ± 0.5	72.2 ± 0.5	79.3 ± 0.4	91.9 ± 0.2	86.9 ± 0.3	68.6 ± 0.2	31.8 ± 0.1	82.5 ± 0.2
–	√	√	81.9 ± 0.4	72.1 ± 0.3	80.3 ± 0.3	82.6 ± 0.3	74.9 ± 0.4	68.0 ± 0.2	31.6 ± 0.2	82.5 ± 0.1
√	√	√	**83.4 ± 0.4**	**73.0 ± 0.3**	**81.9 ± 0.3**	**93.2 ± 0.2**	**88.9 ± 0.3**	**68.8 ± 0.1**	**31.9 ± 0.1**	**82.6 ± 0.1**

Table 3: Classification accuracy (%) of each component in our proposed method on all datasets.

Experiments

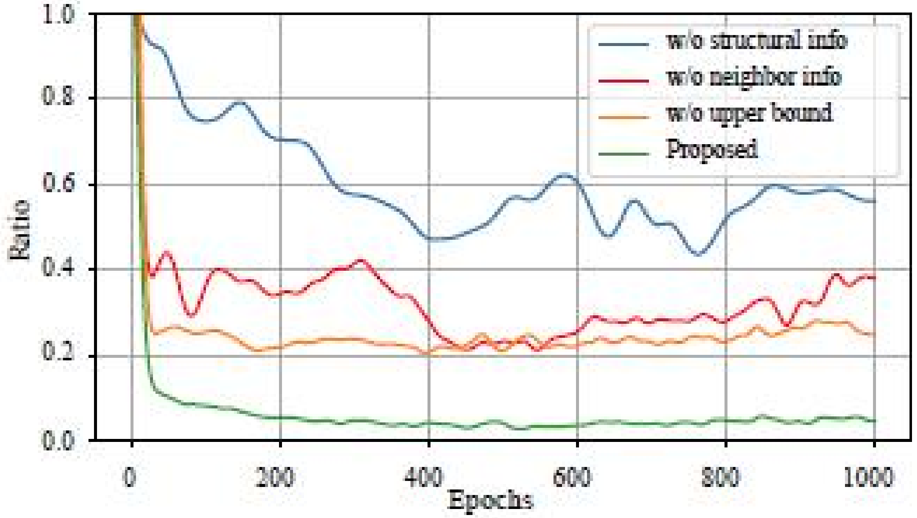

Figure 3: The ratio of intra-class variance to inter-class variation on the dataset Photo.

Experiments

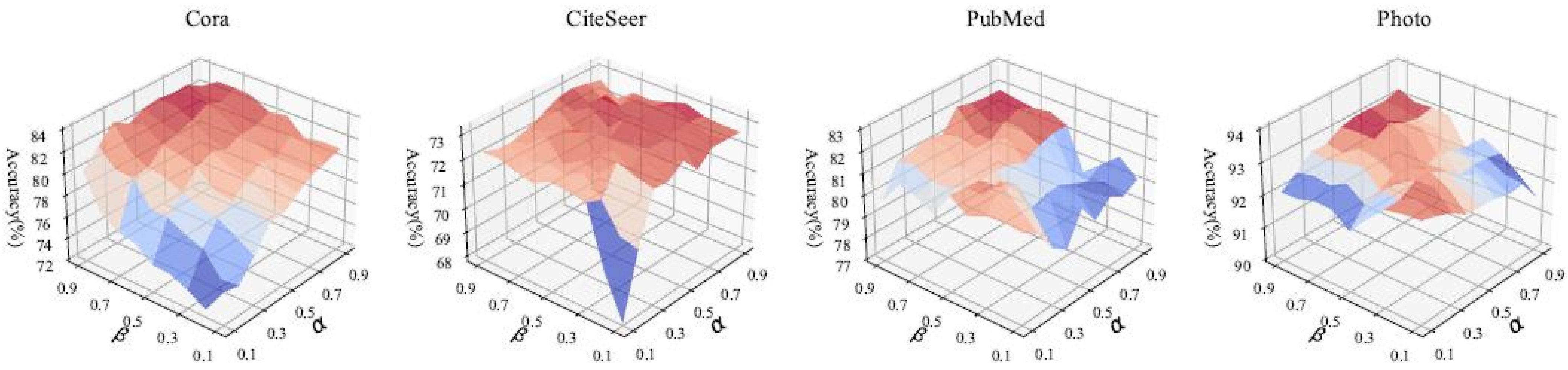

Figure 4: Classification results of our method at different parameter settings (*i.e.*, α and β).

Experiments

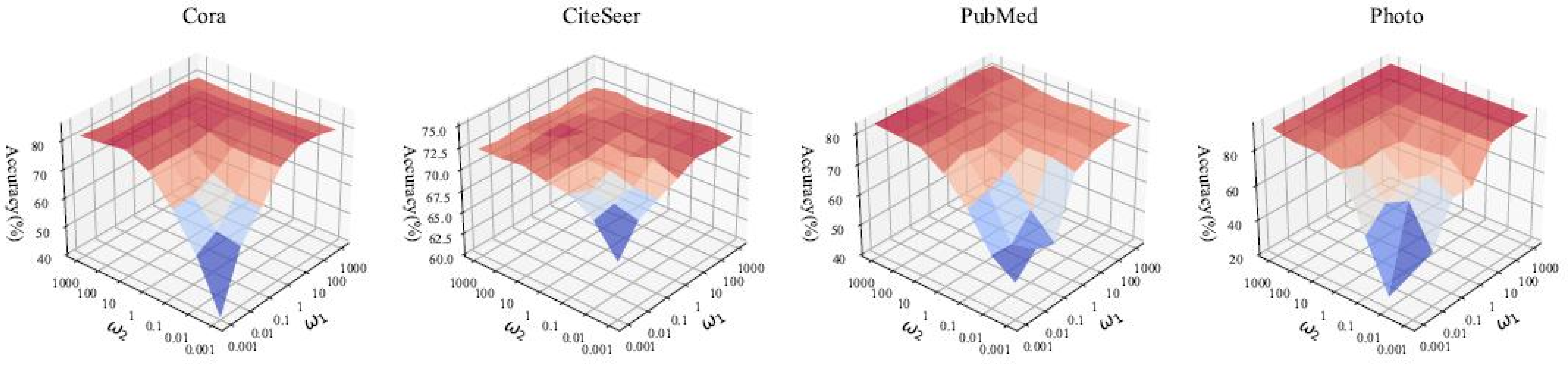

Figure 5: Classification results of our method at different parameter settings (*i.e.*, ω_1 and ω_2).

Experiments

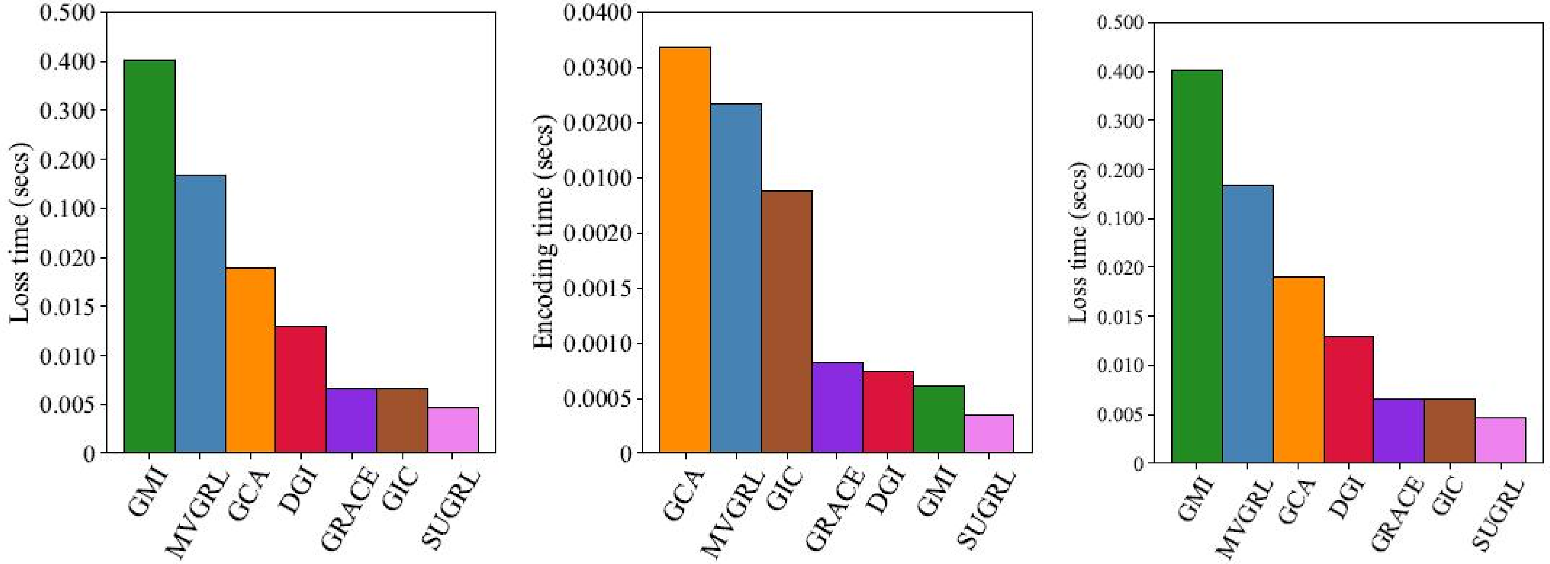

Figure 6: Execution time (seconds) of different parts in all methods on the dataset Photo.

Thank you!