Dual Bidirectional Graph Convolutional Networks for Zero-shot Node Classification（KDD_2022）

Qin Yue
School of Computer and Information Technology, Shanxi University, Taiyuan, China
993203718@qq.com

Jiye Liang*
School of Computer and Information Technology, Shanxi University, Taiyuan, China
ljy@sxu.edu.cn

Junbiao Cui
School of Computer and Information Technology, Shanxi University, Taiyuan, China
945546899@qq.com

Liang Bai
School of Computer and Information Technology, Shanxi University, Taiyuan, China
bailiang@sxu.edu.cn

2022. 8. 31 • ChongQing

gesis Leibniz-Institut für Sozialwissenschaften

Reported by Lele Duan

Code & dataset:https://github.com/warmerspring/DBiGCN

Background

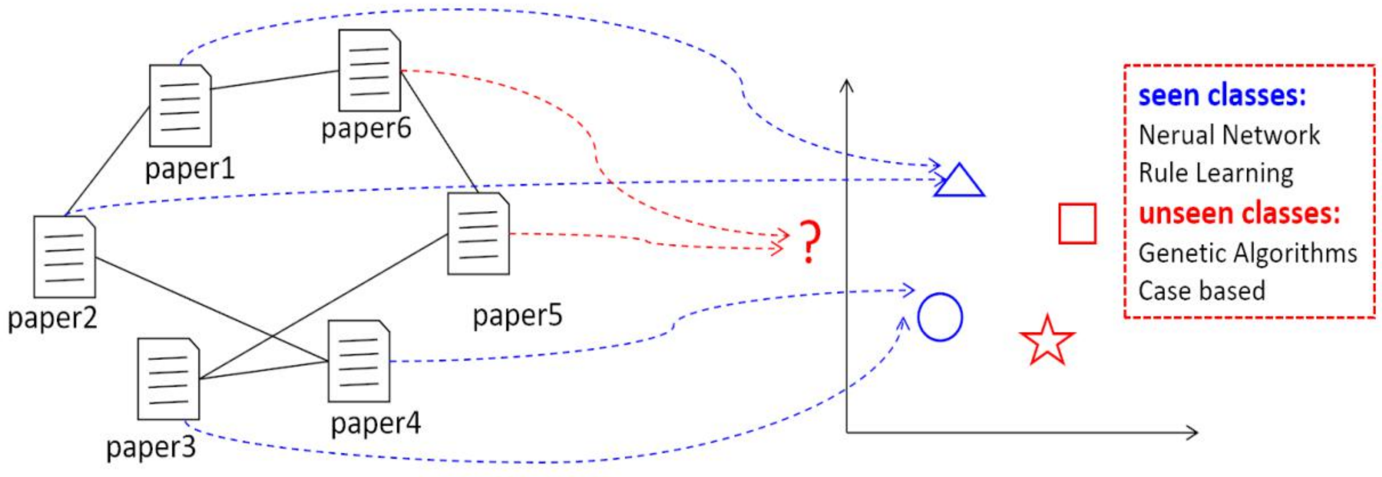


Figure 1: An example of zero-shot node classification.

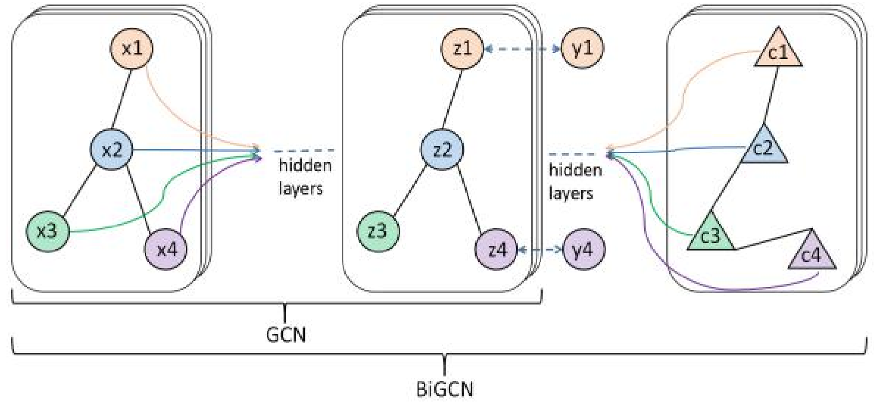


Figure 3: A schematic depiction of BiGCN. The circles represent the nodes and the black lines between the circles represent the relations between the nodes. And the triangles represent the classes and the black lines between the triangles represent the relations between the classes.

- *In order to predict the unlabeled nodes from unseen classes, zero-shot node classification needs to transfer knowledge from seen classes to unseen classes.*
- *However, the GCN only considers the relations between the nodes, not the relations between the classes.*

➢ Over view

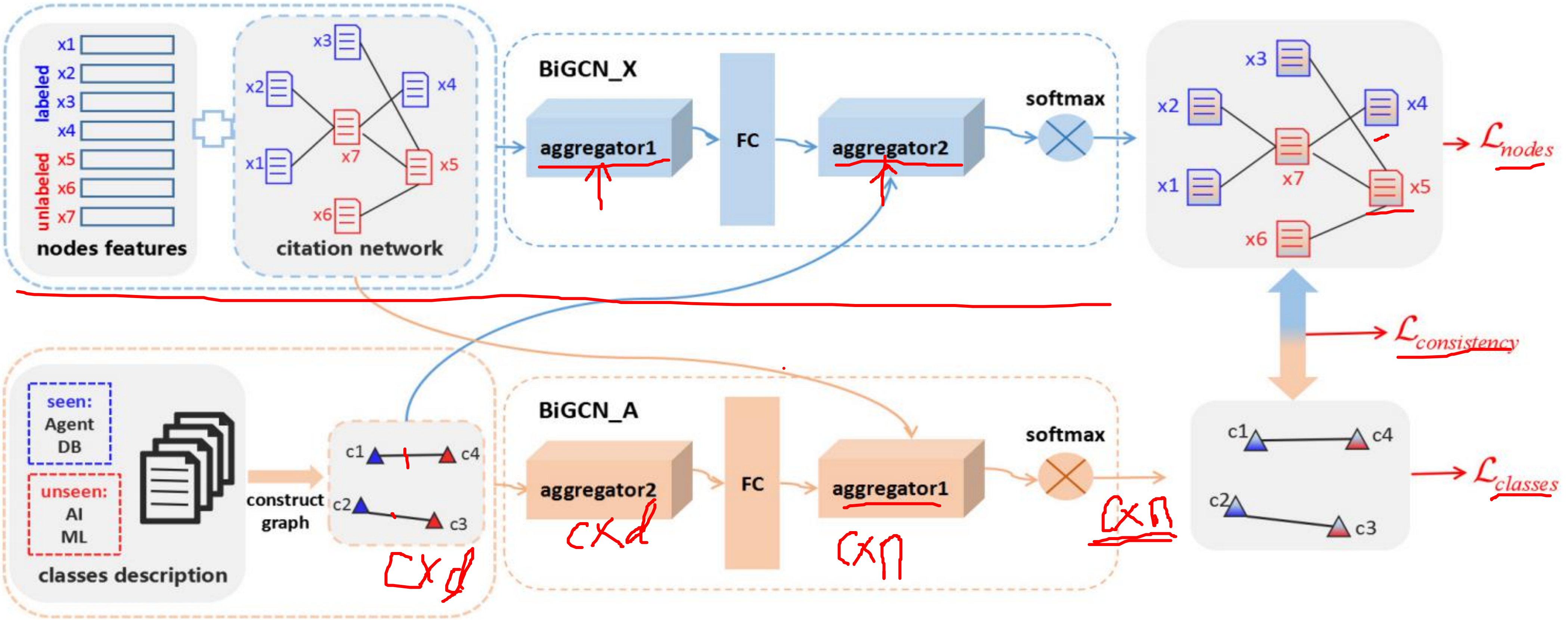


Figure 2: A schematic overview of DBiGCN. The DBiGCN consists of the dual BiGCNs from perspective of the nodes and the classes respectively and the mutual guidance between the dual BiGCNs is achieved via the consistency loss, which is united into a network. The aggregator 1 and 2 are used for aggregating the adjacency information of the nodes and the classes.

Problem Formulation

$G = (V, E, \mathbf{X}, \mathbf{S}^V)$

$V = \{v_1, v_2, ..., v_n\}$

$E \subseteq V \times V$

$\mathbf{S}^V \in \mathbb{R}^{n \times n}$

$\mathbf{X} \in \mathbb{R}^{n \times d}$

$\mathcal{Y} = \mathcal{Y}_s \bigcup \mathcal{Y}_u$ and $\mathcal{Y}_s \bigcap \mathcal{Y}_u = \phi$

c_s seen classes: $\mathcal{Y}_s = \{1, 2, \cdots, c_s\}$

c_u unseen classes: $\mathcal{Y}_u = \{c_s+1, c_s+2, \cdots, c_s+c_u = c\}$.

Each class is described by a semantic description vector $\mathbf{a}_k \in \mathbb{R}^{d_c}$, $k = 1, 2, \cdots, c$ and $\mathbf{A} \in \mathbb{R}^{c \times d_c}$ is the matrix of semantic description vectors of all classes.

Without loss of generality, we assume that the first l nodes are labeled and the rest u nodes are unlabeled and $l + u = n$. All the

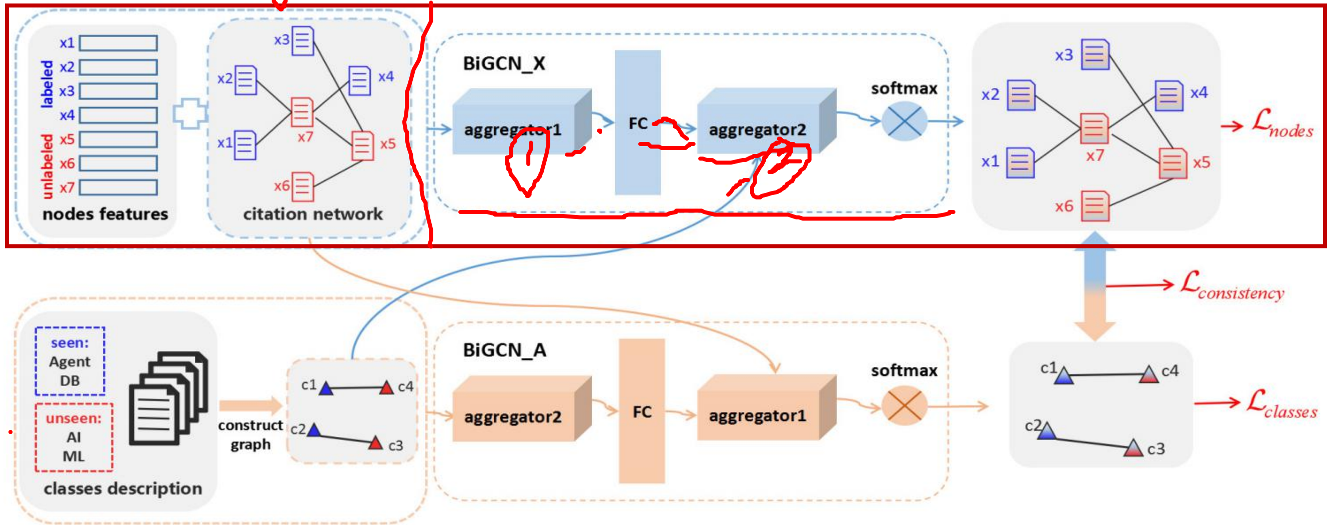
BiGCN_X

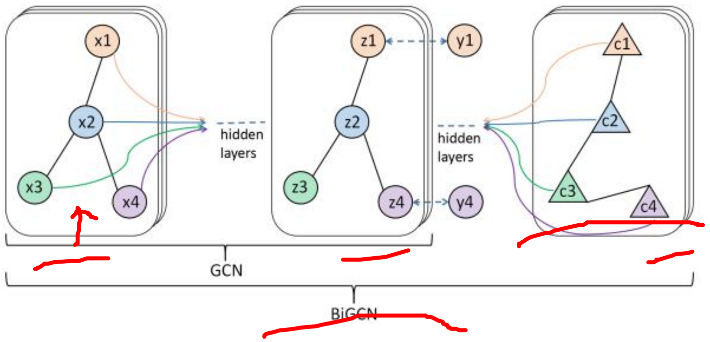
$$\mathbf{Y}^V = \mathrm{softmax}\left(\mathrm{relu}\left(\hat{\mathbf{S}}^V \mathbf{X}\mathbf{W}^{(1)}\right)\mathbf{W}^{(2)}\hat{\mathbf{S}}^{\mathbf{A}}\right), \qquad (4)$$

where $\hat{\mathbf{S}}^{\mathbf{A}}$ is the normalized adjacency matrix of the classes defined by the distances between the classes, which can intuitively reflect the relations between the classes. And $\mathbf{W}^{(1)} \in \mathbb{R}^{d\times d'}$, $\mathbf{W}^{(2)} \in \mathbb{R}^{d'\times c}$ are the learnable parameters. In BiGCN, dimension of predicting

$$\mathcal{L}_{\text{nodes}} = -\sum_{i=1}^{l}\sum_{j=1}^{c} y_{L_{ij}}^{\text{true}} \ln y_{ij}^{V}, \qquad (5)$$

where y_{ij}^{V} is the ith row and jth column entity of the matrix $\mathbf{Y}^V$ and denotes the predicting probability of the ith nodes belonging to class j based on BiGCN from perspective of the nodes. The BiGCN from perspective of the nodes is referenced as BiGCN_X.





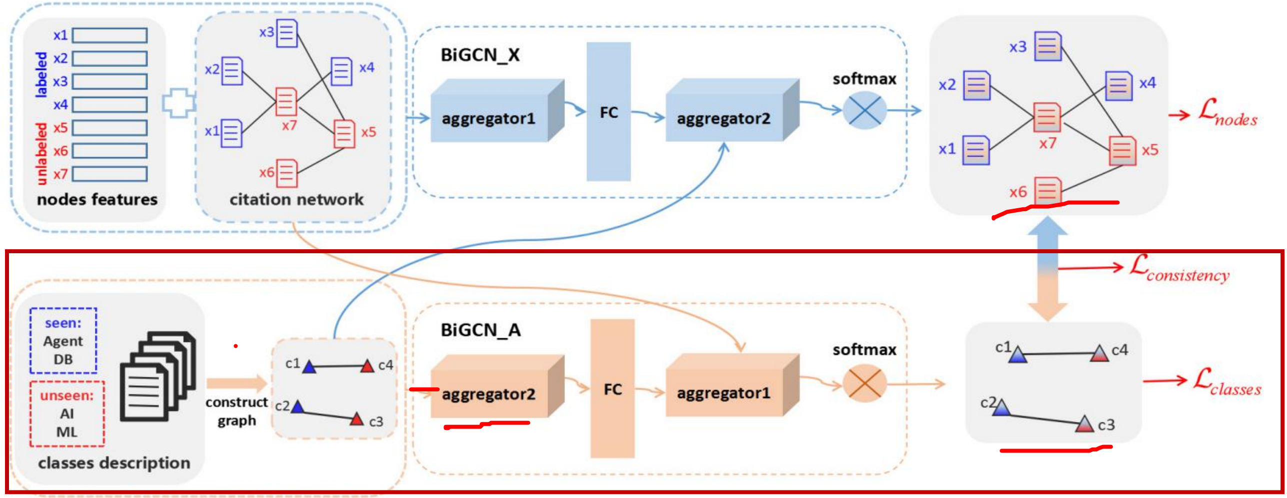
BiGCN_A

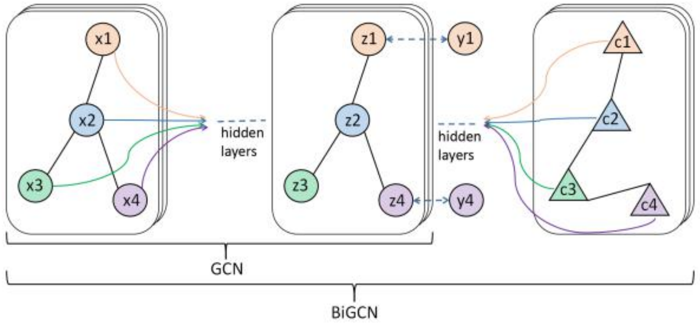
$$\mathbf{Y}^{\mathbf{A}} = \mathrm{softmax}\left(\hat{\mathbf{S}}^{\mathbf{A}}\mathbf{A}\mathbf{W}^{(3)}\hat{\mathbf{S}}^{V}\right), \tag{6}$$

where $\hat{\mathbf{S}}^{\mathbf{A}}$ is the normalized adjacency matrix of the classes that is can be defined by the distances between the classes and $\mathbf{W}^{(3)} \in \mathbb{R}^{d_c \times n}$ is the learnable parameter. The rows of $\mathbf{Y}^{\mathbf{A}} \in \mathbb{R}^{c \times n}$ can be regarded as the representations of the classes, and the columns can be regarded as the representations of the nodes. Finally, the cross-entropy loss function also be applied to all labeled nodes, we have

$$\mathcal{L}_{\text{classes}} = -\sum_{i=1}^{l}\sum_{j=1}^{c} y_{L_{ij}}^{\text{true}} \ln y_{ji}^{\mathbf{A}}, \tag{7}$$

where $y_{ji}^{\mathbf{A}}$ is the jth row and ith column entity of the matrix $\mathbf{Y}^{\mathbf{A}}$ and denotes the predicting probability of the ith nodes belonging to class j based on the BiGCN from perspective of the classes.





Label Consistency Loss

$$\mathcal{L}_{\text{consistency}} = \sum_{i=1}^{l}\sum_{j=1}^{l}\left(\mathbf{y}_i^V \mathbf{y}_j^{\mathbf{A}} - \mathbf{y}_i^{\text{true}}\left(\mathbf{y}_j^{\text{true}}\right)^T\right)^2, \quad (8)$$

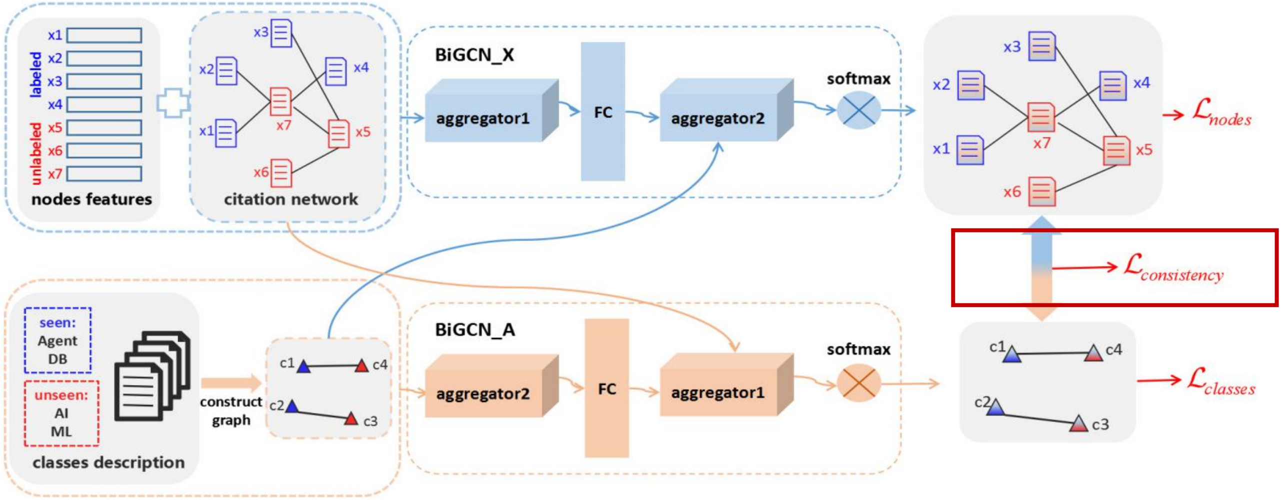
where $\mathbf{y}_i^V \in [0,1]^{1\times c}$ denotes the ith row of the $\mathbf{Y}^V$ and is the predicting label probability vector of the ith nodes based on the BiGCN_X. Similarly, $\mathbf{y}_i^{\mathbf{A}} \in [0,1]^{c\times 1}$ denotes the ith column of the $\mathbf{Y}^{\mathbf{A}}$ and is the predicting label probability vector of the ith nodes based on the BiGCN_A. And $\mathbf{y}_i^{\text{true}}$ is the true one-hot label vector of the ith nodes.

For simplicity, formula (8) can be formulated as

$$\mathcal{L}_{\text{consistency}} = \left\|\mathbf{Y}_L^V \mathbf{Y}_L^{\mathbf{A}} - \mathbf{Y}_L^{\text{true}}\left(\mathbf{Y}_L^{\text{true}}\right)^T\right\|_F^2, \quad (9)$$

where $\mathbf{Y}_L^V \in [0,1]^{l\times c}$ is the predicting label matrix of the l labeled nodes based on the BiGCN_X. Similarly, $\mathbf{Y}_L^{\mathbf{A}} \in [0,1]^{c\times l}$ is the predicting label matrix of the l labeled nodes based on the BiGCN_A. $\mathbf{Y}_L^{\text{true}}$ is the true label matrix of the l labeled nodes.

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{nodes}} + \alpha\mathcal{L}_{\text{classes}} + \beta\mathcal{L}_{\text{consistency}}, \quad (3)$$



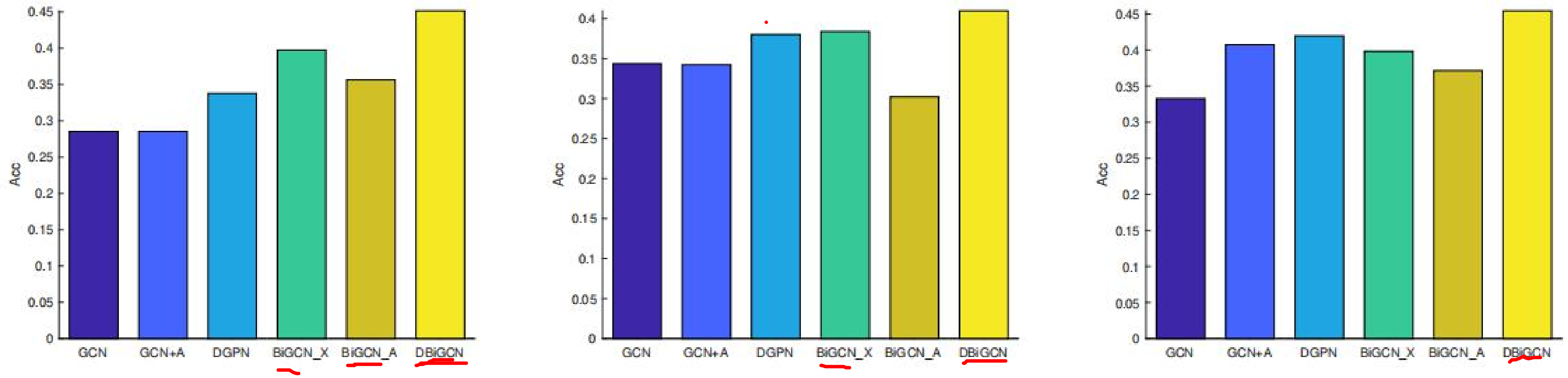
Experiment

Table 3: Zero-shot node classification accuracy (%) using the TEXT-CSDs

		Cora	Citeseer	C-M10M
Class Split I	RandomGuess	25.35	24.86	33.21
	DAP	26.56	34.01[3]	38.71[3]
	DAP(CNN)	27.80	30.45	32.97
	ESZSL	27.35	30.32	37.00
	ZS-GCN	25.73	28.62	37.89
	ZS-GCN(CNN)	16.01	21.18	36.44
	WDVSc	30.62[3]	23.46	38.12
	Hyperbolic-ZSL	26.36	34.18	35.80
	DGPN	33.78[2]	38.02[2]	41.98[2]
	DBiGCN	**45.14**[1]	**40.97**[1]	**45.45**[1]
	Improve rate	33.63%	7.76%	8.27%
Class Split II	RandomGuess	32.69	50.48	49.73
	DAP	30.22	53.30	46.79
	DAP(CNN)	29.83	50.07	46.29
	ESZSL	38.82[3]	55.32[3]	56.07[3]
	ZS-GCN	29.53	52.22	56.07
	ZS-GCN(CNN)	33.20	49.27	51.37
	WDVSc	34.13	52.70	46.26
	Hyperbolic-ZSL	37.02	46.27	55.07
	DGPN	46.40[2]	**61.90**[1]	62.46[2]
	DBiGCN	**49.20**[1]	60.11[2]	**71.86**[1]
	Improve rate	6.03%	-2.89%	15.05%

Table 4: The Comparison of zero-shot node classification accuracy (%) using the different CSDs

		Cora			Citeseer			C-M10M		
		TEXT	LABEL	Decline rate	TEXT	LABEL	Decline rate	TEXT	LABEL	Decline rate
Class Split I	DAP	26.56	25.34	-4.59%	34.01	30.01	-11.76%	38.71	32.67	-15.60%
	ESZSL	27.35	25.79	-5.70%	30.32	28.52	-5.94%	37.00	35.02	-5.35%
	ZS-GCN	25.73	23.73	-7.77%	28.62	26.11	-8.77%	37.89	33.32	-12.06%
	WDVSc	30.62	18.73	-38.83%	23.46	19.70	-16.02%	38.12	30.82	-19.15%
	Hyperbolic-ZSL	26.36	25.47	-3.38%	34.18	21.04	-38.44%	35.80	34.49	-3.66%
	DGPN	33.78	32.55	-3.64%	38.02	31.83	-16.28%	41.98	35.05	-16.51%
	DBiGCN	**45.14**	**39.05**	-13.49%	**40.97**	**39.10**	-3.10%	**45.45**	**43.71**	-3.83%



(a) Cora (b) Citeseer (c) C-M10M

Figure 4: The comparison of the different methods based on Graph Convolutional Network for zero-shot node classification. The abscissa represents the different methods and the ordinate represents the accuracy of the zero-shot node classification.

Experiment

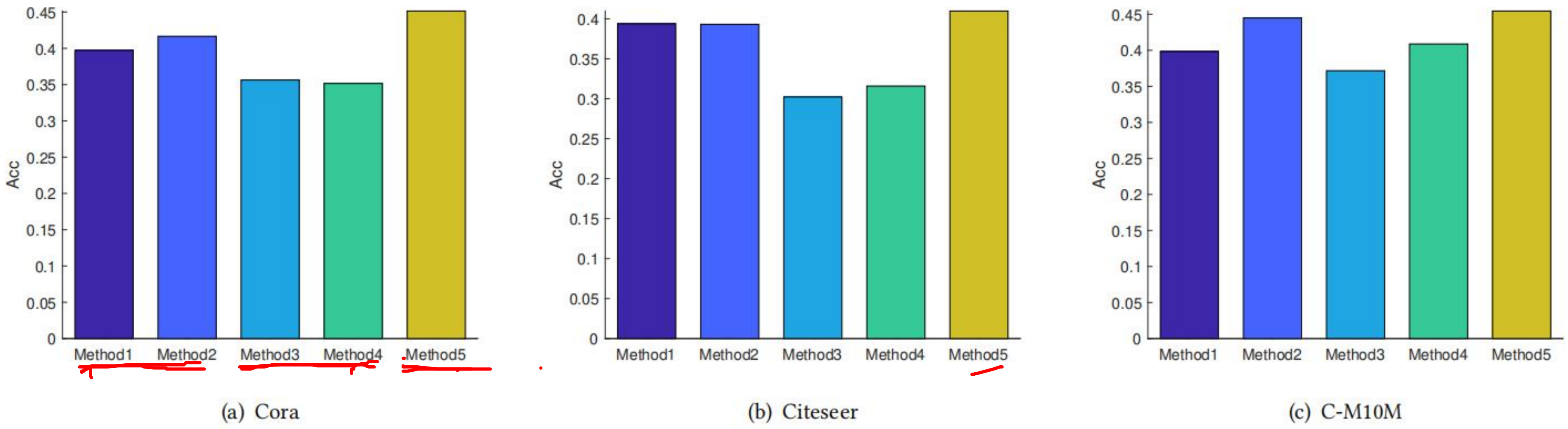


Figure 5: The zero-shot node classification accuracy of the five ablative methods from the proposed model.

Experiment

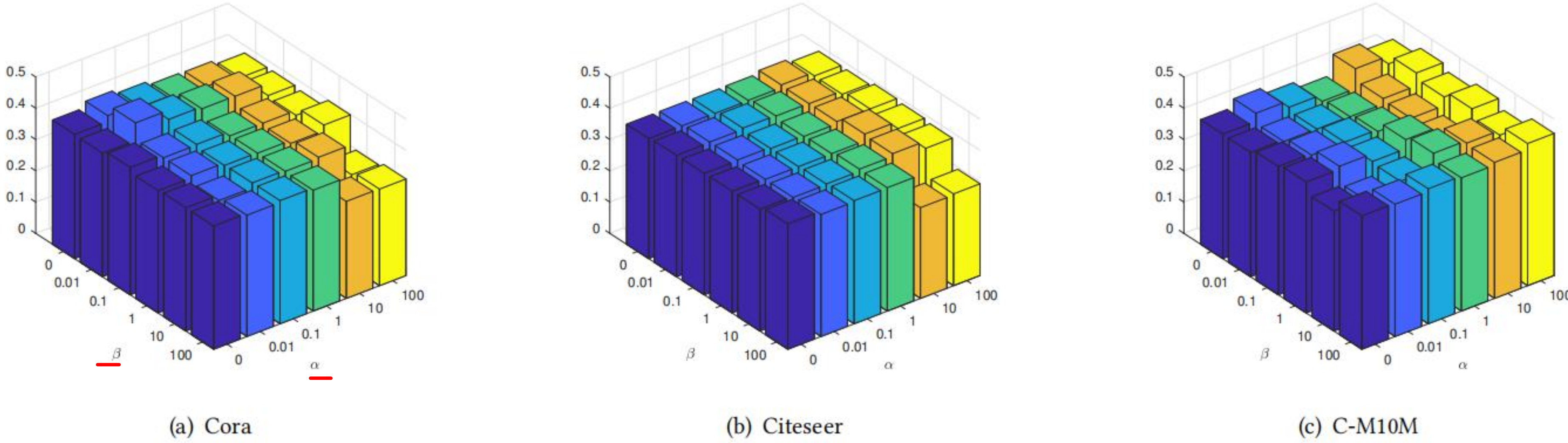


Figure 6: The variations of the zero-shot node classification accuracy of the proposed method under different parameters α and β on all data sets.

Thank you!