Incomplete Graph Representation and Learning via Partial Graph Neural Networks

Bo Jiang
School of Computer Science and Technology
Anhui University
jiangbo@ahu.edu.cn

Ziyan Zhang
School of Computer Science and Technology
Anhui University
zhangziyanahu@163.com

Abstract

Graph Neural Networks (GNNs) are gaining increasing attention on graph data learning tasks in recent years. However, in many applications, graph may be coming in an incomplete form where attributes of graph nodes are partially unknown/missing. Existing GNNs are generally designed on complete graphs which can not deal with attribute-incomplete graph data directly. To address this problem, we develop a novel *partial aggregation* based GNNs, named Partial Graph Neural Networks (PaGNNs), for attribute-incomplete graph representation and learning. Our work is motivated by the observation that the neighborhood aggregation function in standard GNNs can be equivalently viewed as the neighborhood reconstruction formulation. Based on it, we define two novel partial aggregation (reconstruction) functions on incomplete graph and derive PaGNNs for incomplete graph data learning. Extensive experiments on several datasets demonstrate the effectiveness and efficiency of the proposed PaGNNs.

1 Instruction

Learning on graph data is an important problem in machine learning and data mining area. Graph Neural Networks (GNNs) [1, 2, 3, 4, 5, 6] have been demonstrated great success in representation learning of graph data and have gained an increasing attention in recent years. Given an input graph $G(\mathbf{A}, \mathbf{H})$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{H} \in \mathbb{R}^{n \times d}$ denote the adjacency matrix and node attributes respectively, the core aspect of GNNs is their *message propagation scheme* in which each node updates its representation by aggregating the representations from its neighbors via a specific aggregation function [7, 3, 1].
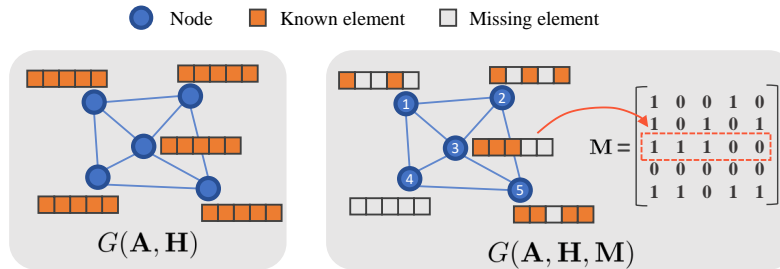

Figure 1: Illustration of complete graph $G(\mathbf{A}, \mathbf{H})$ and attribute-incomplete graph $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$.

However, in many applications, graph data is usually coming in an incomplete form in which attributes of graph nodes are partially missing/unknown [8, 9]. For example, some attributes are unknown or missing due to some difficulties or human errors in data collection process [8]. In this paper, we use

$G(\mathbf{A}, \mathbf{H}, \mathbf{M})$ to represent this incomplete graph where $\mathbf{M} \in \{0,1\}^{n \times d}$ denotes the indicative matrix in which $\mathbf{M}_{ij} = 0$ indicates that $\mathbf{H}_{ij}$ is missing/unknown and $\mathbf{M}_{ij} = 1$ otherwise, as illustrated in Figure 1. Existing GNNs are generally developed on attribute-complete graph data $G(\mathbf{A}, \mathbf{H})$ which can not deal with incomplete graph data $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$ directly.

Related Works. To deal with attribute-incomplete graph $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$, one popular way is to first employ some imputation/completion methods to estimate and fill in the missing values and then adopt traditional GNNs to conduct graph learning. For example, Yoon et al. [10] propose Generative Adversarial Imputation Nets (GAIN) to complete missing data by adapting a GAN [11] framework. Chen et al. [9] recently develop Structure-Attribute Transformer (SAT) for attribute-missing graph completion and then utilize GNNs [1, 4] for incomplete graph node classification. Obviously, this kind of methods conduct missing value estimating and GNN learning independently, which usually leads to weak sub-optimal learning. To overcome this limitation, Taguchi et al. [8] recently propose an end-to-end model GCN_MF which uses a Gaussian mixture model (GMM) to estimate the missing data and jointly learns the GMM and GNN's parameters together. However, GCN_MF [8] increases the amount of training parameters and thus has high computational complexity.

Contributions. Different from previous works, this paper presents a straightforward **partial message-propagation scheme** on incomplete graph to develop novel Partial Graph Neural Networks (PaGNNs) for incomplete graph data learning. Specifically, the proposed partial message propagation is motivated by the observation that the neighborhood aggregation operation in traditional GNNs can be equivalently viewed as the neighborhood reconstruction optimization. Based on it, we derive a partial reconstruction formulation and develop a novel partial message-propagation scheme for GNN's layer-wise propagation. Overall, the main contributions of this paper are summarized as follows,

- We re-interpret neighborhood aggregation in GNNs from neighborhood reconstruction perspective and derive a novel partial aggregation function for GNN's layer-wise message propagation on incomplete graph.

- We develop a simple and effective Partial Graph Neural Networks (PaGNNs) to deal with incomplete graph data representation and learning.

Experiments on several datasets demonstrate the effectiveness and efficiency of the proposed PaGNNs.

2 Revisiting GNNs

It is known that the core aspect of GNNs is their *message propagation scheme* in which each node updates its embedding by aggregating the embeddings from its neighbors via a specific aggregation function [7, 3, 1, 12, 13]. Given an input graph $G(\mathbf{A}, \mathbf{H})$ where $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_n) \in \mathbb{R}^{n \times d}$, the message propagation for layer-wise propagation in GNNs can generally be formulated as,

$$\mathbf{h}_i' \leftarrow \sigma\big(\phi_{agg}\left(\mathbf{A}_{ij}, \mathbf{h}_j, j \in \mathcal{N}_i\right)\mathbf{W}\big) \tag{1}$$

where ϕ_{agg} denotes the specific aggregation function and $\mathcal{N}_i$ denotes the i-th node's neighborhood. $\mathbf{W}$ is network's weight parameter and $\sigma(\cdot)$ denotes an activation function. For example, one popular way to define the ϕ_{agg} is using weighted mean function [3] as

$$\phi_{agg_m}\left(\mathbf{A}_{ij}, \mathbf{h}_j, j \in \mathcal{N}_i\right) = \frac{1}{\tilde{\mathbf{d}}_i} \sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij}\mathbf{h}_j \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\tilde{\mathbf{d}}_i = \sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij}$ and $\mathbf{I}$ is an identity matrix. In addition, Kipf et al. [1] propose a commonly used Graph Convolutional Networks (GCN) which defines ϕ_{agg} as a symmetric normalized summation function as

$$\phi_{agg_n}\left(\mathbf{A}_{ij}, \mathbf{h}_j, j \in \mathcal{N}_i\right) = \sum_{j \in \mathcal{N}_i} \frac{1}{\sqrt{\tilde{\mathbf{d}}_i \tilde{\mathbf{d}}_j}} \tilde{\mathbf{A}}_{ij}\mathbf{h}_j \tag{3}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{d}}_i = \sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij}$.

3 Partial Aggregation for Incomplete Graph Neural Networks

The above standard aggregation functions $(\phi_{agg_m}, \phi_{agg_n})$ and message propagation scheme are defined on complete graph $G(\mathbf{A}, \mathbf{H})$, which limit standard GNNs [3, 1] to deal with incomplete graph data $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$. To overcome this problem, in this section, we define a novel *partial* aggregation and message propagation scheme on incomplete graph $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$ and propose Partial Graph Neural Networks (PaGNNs) for incomplete graph data learning.

Our partial aggregations are motivated based on re-formulations of the above neighborhood aggregations $(\phi_{agg_m}, \phi_{agg_n})$ from a neighborhood reconstruction perspective. Specifically, one can easily prove that Eq.(2) provides the optimal solution to the following neighborhood reconstruction problem,

$$\phi_{agg_m}\left(\mathbf{A}_{ij}, \mathbf{h}_j, j \in \mathcal{N}_i\right) = \arg\min_{\mathbf{z}} \sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij} \|\mathbf{z} - \mathbf{h}_j\|^2 \tag{4}$$

where $\|\cdot\|$ denotes Frobenius norm [7] and $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$.

Remark. Eq.(4) provides a different view of the aggregation operation Eq.(2) in GNN, i.e., each node aims to reconstruct its representation from its neighbors' representations.

Using this formulation, we can derive a simple partial reconstruction (aggregation) function on incomplete graph $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$ as

$$\phi_{pagg_m}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right) = \arg\min_{\mathbf{z}} \sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij} \|\mathbf{m}_j \odot (\mathbf{z} - \mathbf{h}_j)\|^2 \tag{5}$$

where $\mathbf{m}_j$ denotes the j-th row of indicative matrix $\mathbf{M}$ and $\odot$ denotes Hadamard product operation. Moreover, Eq.(5) also has a simple closed-form solution which is computed by setting the first derivation w.r.t $\mathbf{z}$ to $\mathbf{0}$ and obtain

$$\phi_{pagg_m}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right) = \frac{\sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij}(\mathbf{m}_j \odot \mathbf{h}_j)}{\sum_{j \in \mathcal{N}_i} \tilde{\mathbf{A}}_{ij}\mathbf{m}_j} \tag{6}$$

Note that Eq.(6) provides a partial neighborhood aggregation on (attribute) incomplete graph. Using this partial aggregation function ϕ_{pagg_m}, we can define a *message propagation scheme* on incomplete graph as

$$\mathbf{h}'_i \leftarrow \sigma\big(\phi_{pagg_m}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right)\mathbf{W}\big) \tag{7}$$

Similarly, we can further prove that Eq.(3) provides the optimal solution to problem as

$$\phi_{agg_n}\left(\mathbf{A}_{ij}, \mathbf{h}_j, j \in \mathcal{N}_i\right) = \arg\min_{\mathbf{z}} \sum_{j \in \mathcal{N}_i} \hat{\mathbf{A}}_{ij} \left\|\frac{\mathbf{z}}{\hat{\mathbf{d}}_i} - \mathbf{h}_j\right\|^2 \tag{8}$$

where $\hat{\mathbf{d}}_i = \sum_j \hat{\mathbf{A}}_{ij}$, $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ and $\tilde{\mathbf{D}}$ is the diagonal matrix with $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. Using this formulation, we can also derive a symmetric normalized type partial aggregation function on incomplete graph $G(\mathbf{A}, \mathbf{H}, \mathbf{M})$ as

$$\phi_{pagg_n}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right) = \arg\min_{\mathbf{z}} \sum_{j \in \mathcal{N}_i} \hat{\mathbf{A}}_{ij} \left\|\mathbf{m}_j \odot \left(\frac{\mathbf{z}}{\hat{\mathbf{d}}_i} - \mathbf{h}_j\right)\right\|^2 \tag{9}$$

Eq.(9) also has a simple closed-form solution as

$$\phi_{pagg_n}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right) = \frac{\hat{\mathbf{d}}_i \sum_{j \in \mathcal{N}_i} \hat{\mathbf{A}}_{ij}(\mathbf{m}_j \odot \mathbf{h}_j)}{\sum_{j \in \mathcal{N}_i} \hat{\mathbf{A}}_{ij}\mathbf{m}_j} \tag{10}$$

Similar to Eq.(7), using this partial aggregation function ϕ_{pagg_n}, we can define a normalized-type message propagation scheme on incomplete graph as

$$\mathbf{h}'_i \leftarrow \sigma\big(\phi_{pagg_n}\left(\mathbf{A}_{ij}, \mathbf{h}_j, \mathbf{m}_j, j \in \mathcal{N}_i\right)\mathbf{W}\big) \tag{11}$$

Remark. Using the above message propagation operations (ϕ_{pagg_m} and ϕ_{pagg_n}), we can develop two novel multi-layer GNNs for incomplete graph data learning. In this paper, we call them as Partial

Graph Neural Networks (PaGNNs). PaGNNs (PaGNN-M, PaGNN-N) have two main advantages. First, PaGNNs adopt standard GNNs-like message propagation scheme and thus can be implemented as simply and efficiently as standard GNNs. In particular, both Eq.(7) and Eq.(11) have GCN-like [1] matrix formulation as

$$\mathbf{H}' \leftarrow \sigma\big[(\tilde{\mathbf{A}}(\mathbf{M} \odot \mathbf{H}) \oslash \tilde{\mathbf{A}}\mathbf{M})\mathbf{W}\big] \tag{12}$$

$$\mathbf{H}' \leftarrow \sigma\big[(\hat{\mathbf{A}}\mathbf{E} \odot \hat{\mathbf{A}}(\mathbf{M} \odot \mathbf{H}) \oslash \hat{\mathbf{A}}\mathbf{M})\mathbf{W}\big] \tag{13}$$

where $\odot$ and $\oslash$ denote the Hadamard product and element division operation respectively and $\mathbf{E} \in \mathbb{R}^{n \times d}$ with all elements $\mathbf{E}_{ij} = 1$. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ and $\tilde{\mathbf{D}}$ is the diagonal matrix with $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. Second, PaGNNs conduct on incomplete graph directly and do not need some additional parameters to estimate missing values. This is the key difference between our PaGNNs and previous related works [10, 8, 9].

The proposed PaGNNs are general which can be potentially used in many incomplete graph learning tasks, such as graph classification [14], link prediction [15] and clustering [16]. In this paper, we mainly focus on semi-supervised node classification [1]. Figure 2 shows the cross-entropy classification loss of PaGNNs across different epochs by using well known Adam algorithm [17] on Cora[18] dataset with 50% unknown/missing attributes (as introduced in Experiments in detail). It illustrates the convergence of the proposed PaGNNs (PaGNN-M (Eq.(12)) and PaGNN-N (Eq.(13))).
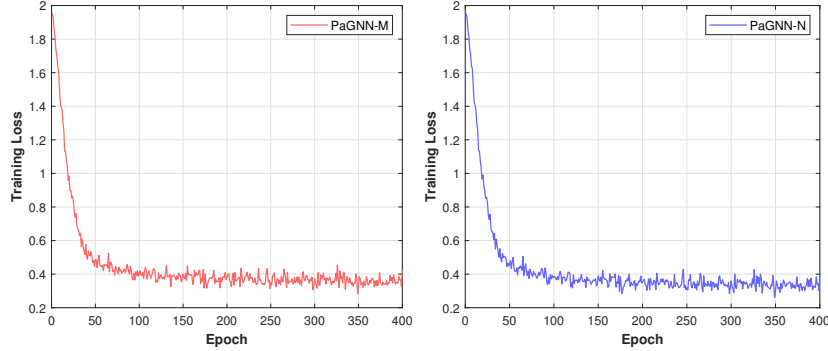

Figure 2: Demonstration of training loss values across different epochs on Cora dataset with 50% unknown/missing attributes.

4   Experiment

We evaluate the effectiveness and efficiency of the proposed PaGNNs on four datasets and compare them against several related approaches.

Table 1: Statistics of datasets.

	#Nodes	#Edges	#Attributes	#Classes
Cora [18]	2708	5429	1433	7
Citeseer [18]	3327	4732	3703	6
AmaPhoto [19]	7650	143663	745	8
AmaComp [19]	13752	287209	767	10

4.1   Experimental Setup

We test the proposed PaGNNs on several commonly used datasets including Cora [18], Citeseer [18], AmaPhoto [19] and AmaComp [19]. The introductions of these datasets are summarized in Table 1. For Cora and Citeseer datasets, similar to the standard data splits [1, 8], we randomly choose 20
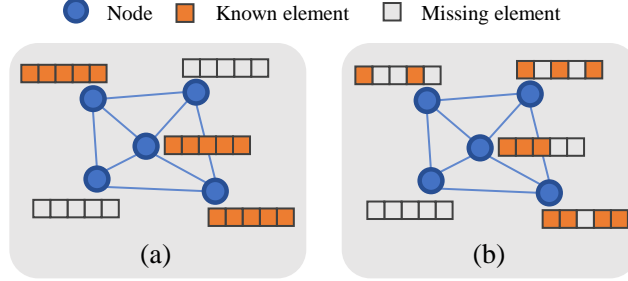
4

Figure 3: The illustration of two types of incomplete graph.

nodes per each class for training, 500 nodes for validation, 1000 nodes for testing. For AmaPhoto and AmaComp datasets, we randomly select 40 nodes per each class for training, 500 nodes for validation and the remaining nodes for testing, as used in works [8]. For each dataset, we generate two types of attribute incomplete graph data as follows,

- Type 1: We randomly select p_m percentage of graph nodes and set all attributes of these nodes to unknown/missing values, as illustrated in Figure 3 (a).
- Type 2: We randomly select and set p_m percentage of elements in attribute matrix $\mathbf{H}$ to unknown/missing elements, as illustrated in in Figure 3 (b).

Similar to work [8], for each missing rate p_m, we generate five different incomplete graphs and report the average learning accuracy. For all compared methods, we report the average performance under 20 different network initializations, as suggested in work [8]. Similar to the setting of standard GCN [1], we use a two-layer architecture in our PaGNNs. For node classification task, we optimize all network parameters by minimizing the cross-entropy loss function [1]. The learning and dropout rates of PaGNNs are set to 0.005 and 0.5, respectively. The number of hidden units is set to 16 on Cora and Citeseer datasets and 64 on AmaPhoto and AmaComp datasets. All network parameters are optimized by using Adam algorithm[17] in which the training is stopped if the validation loss does not decrease for 100 epochs, as utilized in many previous works [8, 1].

4.2 Comparison Methods

We first compare our PaGNNs with some commonly used imputation based methods which include Mean Filling (MF) [20], VAE [21] and GAIN [10]. These methods first estimate and fill in the missing values for the incomplete graph and then use GCN [1] on the 'filled' complete graph. We also compare PaGNNs against some recent competitive methods including Structure-Attribute Transformer (SAT)[9] and GCN$_{MF}$ [8]. The detail of these comparison methods are introduced below.

- MF [20] fills in all the missing values in attributes $\mathbf{H}$ by using the mean value of the known elements across different nodes.
- VAE [21] utilizes a VAE model to recover the missing values in attributes $\mathbf{H}$.
- GAIN [10] uses a generative adversarial network (GAN) [11] technique to generate the missing values in attributes $\mathbf{H}$.
- SAT (GCN) [9] first estimates the missing attributes via a GNN-based distribution matching model and then uses standard GNNs [1, 4] for incomplete graph data learning.
- GCN$_{MF}$ [8] employs a learnable Gaussian mixture model (GMM) to fill in the missing data and learns all parameters in an end-to-end manner.

4.3 Comparison Results

Effectiveness Evaluation. Table 2 and Table 3 summarize the comparison results on problem Type 1 and Type 2, respectively. Since SAT [9] can only deal with problem Type 1, we have not compared it in Table 3. Overall, we can note that PaGNNs generally perform better than the other comparison methods, especially on the incomplete graph data with large number of missing values. Specifically,

Table 2: Comparison results on Type 1 problem under different missing rates. * means that the results are obtained from previous work [8].

Method	Cora					Citeseer				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
MF[20]	81.03	79.53	76.67	67.80	35.13	70.27	68.87	63.23	55.67	27.30
VAE*[21]	80.63	78.57	74.69	60.71	17.27	69.63	66.34	60.46	40.71	17.20
GAIN*[10]	80.53	78.36	74.25	61.33	18.43	69.47	65.88	59.96	41.21	17.89
SAT[9]	81.14	80.11	76.64	71.48	51.67	70.33	69.08	65.12	**58.52**	34.32
GCN$_{MF}$*[8]	81.65	80.67	77.43	**72.69**	55.64	70.44	66.57	63.44	56.88	39.86
PaGNN-M	81.40	80.52	**77.54**	68.94	48.78	70.84	68.84	65.02	56.86	41.14
PaGNN-N	**81.78**	**80.82**	77.46	72.32	**56.60**	**71.44**	**69.84**	**66.56**	58.34	**42.42**

Method	AmaPhoto					AmaComp				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
MF[20]	91.77	91.07	90.03	88.87	81.53	84.17	83.50	82.27	79.00	51.40
VAE*[21]	92.11	91.50	90.46	87.47	67.85	82.76	81.72	79.23	73.76	41.37
GAIN*[10]	92.04	91.49	90.63	88.60	76.48	82.76	82.11	80.76	74.38	54.20
SAT[9]	92.10	91.03	90.45	89.99	88.98	83.48	82.66	82.40	80.62	80.17
GCN$_{MF}$*[8]	92.45	92.08	91.52	90.39	86.09	86.37	85.80	85.24	84.06	73.42
PaGNN-M	92.48	92.24	91.70	91.12	89.08	**86.76**	86.26	**85.86**	85.04	82.70
PaGNN-N	**92.74**	**92.42**	**91.96**	**91.52**	**89.12**	86.40	**86.28**	85.84	**85.36**	**83.16**

Table 3: Comparison results on Type 2 problem under different missing rates. * means that the results are obtained from previous work [8].

Method	Cora					Citeseer				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
MF[20]	80.40	78.07	74.37	66.70	37.40	70.87	69.90	66.27	59.83	39.17
VAE*[21]	80.91	79.18	76.84	50.79	13.27	69.80	68.54	50.91	18.45	11.00
GAIN*[10]	80.43	78.35	75.31	70.34	58.87	69.64	67.56	63.86	55.77	42.73
GCN$_{MF}$*[8]	81.70	80.41	77.91	74.38	63.49	70.93	69.84	67.03	60.70	47.78
PaGNN-M	82.02	80.56	77.96	74.90	65.16	70.84	69.88	67.20	61.34	48.50
PaGNN-N	**82.22**	**80.64**	**78.88**	**75.40**	**66.88**	**71.38**	**69.98**	**67.46**	**63.88**	**52.46**

| Method	AmaPhoto					AmaComp				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
MF[20]	91.83	91.50	90.77	89.47	79.83	83.97	83.60	82.67	81.23	62.40
VAE*[21]	92.20	91.90	91.15	89.28	81.43	82.65	81.72	80.47	78.55	67.26
GAIN*[10]	92.23	91.90	91.49	90.72	86.96	82.94	82.44	81.56	79.96	76.15
GCN$_{MF}$*[8]	92.54	92.20	92.09	91.25	88.96	86.32	85.98	85.46	84.03	77.52
PaGNN-M	92.64	92.48	92.14	91.74	90.48	**86.80**	**86.54**	**86.14**	85.32	83.88
PaGNN-N	**92.78**	**92.80**	**92.40**	**91.98**	**90.82**	86.38	86.10	85.78	**85.46**	**84.52**
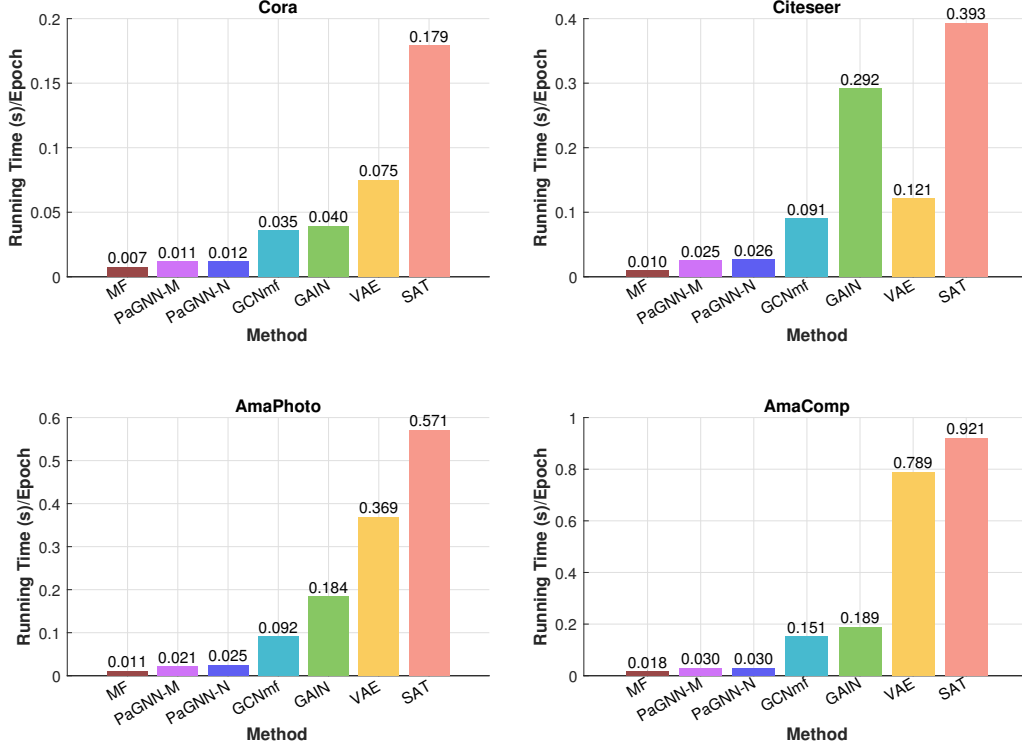
Figure 4: The empirical running time of each epoch for different comparison methods.

we can observe that, (1) PaGNNs perform better than some traditional imputation based methods including MF [20], VAE [21] and GAIN [10] on both types of incomplete graph data. (2) PaGNNs generally obtain better performance than recent SAT [9] method for Type 1 problem. (3) PaGNNs outperform recent GCN$_{MF}$ [8] method which also provides an end-to-end model for incomplete graph data learning. Comparing with GCN$_{MF}$ [8], PaGNNs deal with incomplete graph directly without needing some additional parameters to estimate the unknown information and thus perform more optimally. (4) PaGNN-N generally performs slightly better than PaGNN-M, indicating the more effectiveness of the proposed normalized aggregation function ϕ_{pagg_n} in Eq.(10)

Efficiency Evaluation. Figure 4 shows the running time of each training epoch of PaGNNs on different datasets which are implemented on the computer with NVIDIA GeForce GTX 1080 Ti GPU. For fair comparison, the average running time of each compared two-stage method (MF [20], GAIN [10] , VAE [21], SAT[9]) is calculated as the whole time cost (missing value estimation + GCN training) divided by total epochs on GCN training. Here, we can observe that (1) PaGNNs are slightly slower than MF [20] method which uses a simple mean strategy for data imputation and standard GCN [1] for graph learning. As shown in Eqs.(12,13), PaGNNs have GCN-like message propagation operation and thus can have similar time complexity (slightly higher) with standard GCN [1]. (2) PaGNNs perform obviously faster than other comparing methods including GAIN [10] , VAE [21], SAT[9] and GCN$_{MF}$ [8]. This clearly demonstrates the efficiency of the proposed PaGNNs.

5 Conclusion

In this paper, we develop a novel PaGNNs for attribute-incomplete graph data learning and representation. PaGNNs adopt GCN-like message propagation scheme and thus can be implemented as simply and efficiently as standard GNNs. PaGNNs conduct on incomplete graph directly and do not need some additional parameters to estimate missing values. Extensive experiments show the superiority of our PaGNNs against other comparison methods.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[2] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5115–5124, 2017.

[3] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1024–1034, 2017.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[5] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *International Conference on Learning Representations (ICLR)*, 2019.

[6] Z. Xinyi and L. Chen, "Capsule graph neural network," in *International Conference on Learning Representations (ICLR)*, 2018.

[7] S. Geisler, D. Zügner, and S. Günnemann, "Reliable graph neural networks via robust aggregation," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 13272–13284, 2020.

[8] H. Taguchi, X. Liu, and T. Murata, "Graph convolutional networks for graphs containing missing features," *Future Generation Computer Systems*, vol. 117, pp. 155–168, 2021.

[9] X. Chen, S. Chen, J. Yao, H. Zheng, Y. Zhang, and I. W. Tsang, "Learning on attribute-missing graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International Conference on Machine Learning (ICML)*, pp. 5689–5698, PMLR, 2018.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672–2680, 2014.

[12] L. Zhang and H. Lu, "A feature-importance-aware and robust aggregator for gcn," in *ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 1813–1822, 2020.

[13] L. Yang, F. Wu, Y. Wang, J. Gu, and Y. Guo, "Masked graph convolutional network.," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4070–4077, 2019.

[14] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 4805–4815, 2018.

[15] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *Advances in Neural Information Processing Systems (NIPS) Bayesian Deep Learning Workshop*, 2016.

[16] J. Xie, G. Ross, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning (ICML)*, pp. 478–487, 2016.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[18] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[19] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Special Interest Group on Information Retrieval (SIGIR)*, pp. 43–52, 2015.

[20] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel k-means with incomplete kernels," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[21] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.