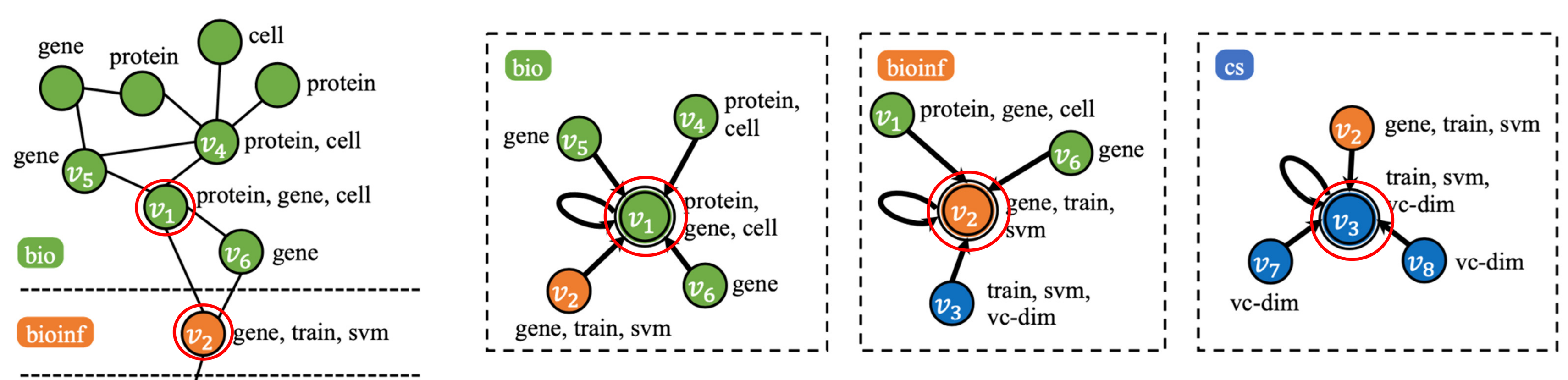
SMU SINGAPORE MANAGEMENT UNIVERSITY
School of Information Systems

Node-wise Localization of Graph Neural Networks

Zemin Liu, Yuan Fang, Chenghao Liu, Steven C.H. Hoi

In Proceeding of 30th International Joint Conference on Artificial Intelligence (IJCAI-21)
21st -26th August, 2021

Motivation

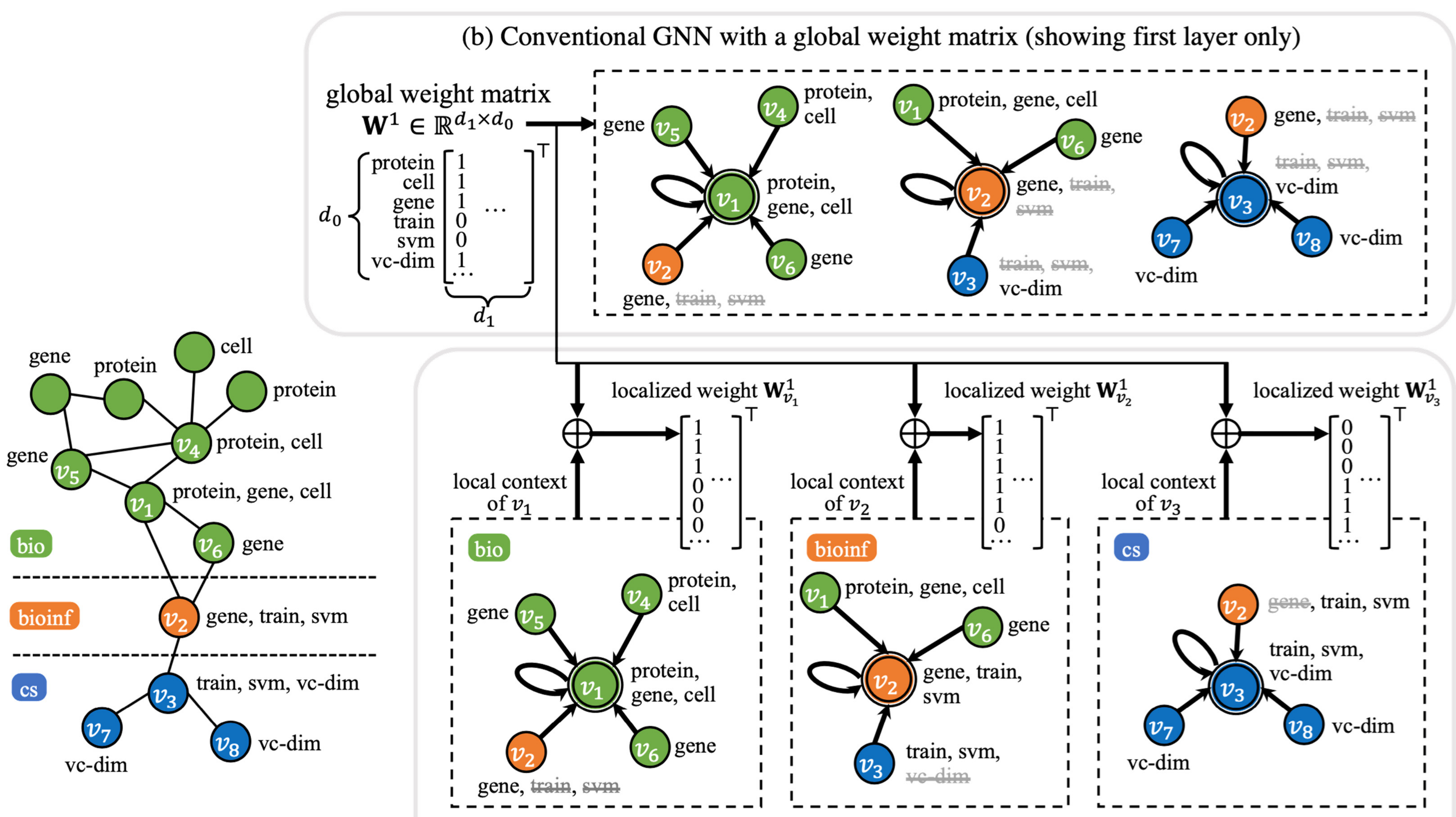


- Different local context of each node
  - Biology: v_1
  - Bioinformatics: v_2
  - Computer science: v_3

Q: Can we allow each node to be parameterized by its own weight matrix?

Localization: Localize the global GNN model for each node

The proposed model: LGNN



(a) Toy citation graph

(b) Conventional GNN with a global weight matrix (showing first layer only)

(c) Node-wise localization with context-dependent weights (showing first layer only)

Consider localization from two aspects: **node-level** and **edge-level**

General formulation of localization:

- Localized model

$$\Theta_v = f(\Theta, C_v)$$

Localized model on v — Global model — Local context of v

- Local context of node v on graph $G = (V, E)$

$$C_v = \{v\} \cup \{u \in V : \langle v, u \rangle \in E\}$$

Node-level localization:

$$\mathbf{W}^l_v = \mathbf{W}^l \odot \left[(\mathbf{a}^l_v)_{\times d_l}\right]^\top + \left[(\mathbf{b}^l_v)_{\times d_l}\right]^\top$$ Localize the weight matrix

$$\mathbf{a}^l_v = \sigma\left(\mathbf{M}^l_a \mathbf{c}^l_v\right) + \mathbf{1}, \quad \mathbf{b}^l_v = \sigma\left(\mathbf{M}^l_b \mathbf{c}^l_v\right)$$ Scaling and shifting factors

$$\mathbf{c}^l_v = \textsc{Mean}\left(\{\mathbf{h}^{l-1}_u : \forall u \in C_v\}\right)$$ Local context

Edge-level localization:

$$\mathbf{h}^l_v = \sigma\left(\textsc{Aggr}\left(\{\mathbf{W}^l_v \mathbf{h}^{l-1}_u \odot \mathbf{a}^l_{u,v} + \mathbf{b}^l_{u,v} : \forall u \in C_v\}\right)\right)$$ Aggregation

$$\mathbf{a}^l_{u,v} = \sigma\left(\mathbf{N}^l_a \mathbf{c}^l_{u,v}\right) + \mathbf{1}, \quad \mathbf{b}^l_{u,v} = \sigma\left(\mathbf{N}^l_b \mathbf{c}^l_{u,v}\right)$$ Scaling and shifting factors

$$\mathbf{c}^l_{u,v} = \textsc{Concat}\left(\mathbf{h}^{l-1}_v, \mathbf{h}^{l-1}_u\right)$$ Local context

Overall loss:

Prediction for semi-supervised node classification

$$\mathbf{z}_{v,k} = \textsc{Softmax}\left(\mathbf{h}^\ell_{v,k}\right) = \frac{\exp\left(\mathbf{h}^\ell_{v,k}\right)}{\sum_{k'=1}^{K} \exp\left(\mathbf{h}^\ell_{v,k'}\right)}$$

Overall loss

Parameters set of global GNN — Parameters set of localization

$$-\sum_{v \in V_Y} \sum_{k=1}^{K} Y_{v,k} \ln \mathbf{z}_{v,k} + \lambda_G \|\Theta_G\|_2^2 + \lambda_L \|\Theta_L\|_2^2$$
$$+\lambda\left(\|A - 1\|_2^2/|A| + \|B\|_2^2/|B|\right)$$

Contains all scaling factors — Contains all shifting factors

Experiments

Experimental setup

Dataset	# Nodes	# Edges	# Classes	# Features
Cora	2,708	5,429	7	1,433
Citeseer	3,327	4,732	6	3,703
Amazon	13,381	245,778	10	767
Chameleon	2,277	36,101	5	2,325

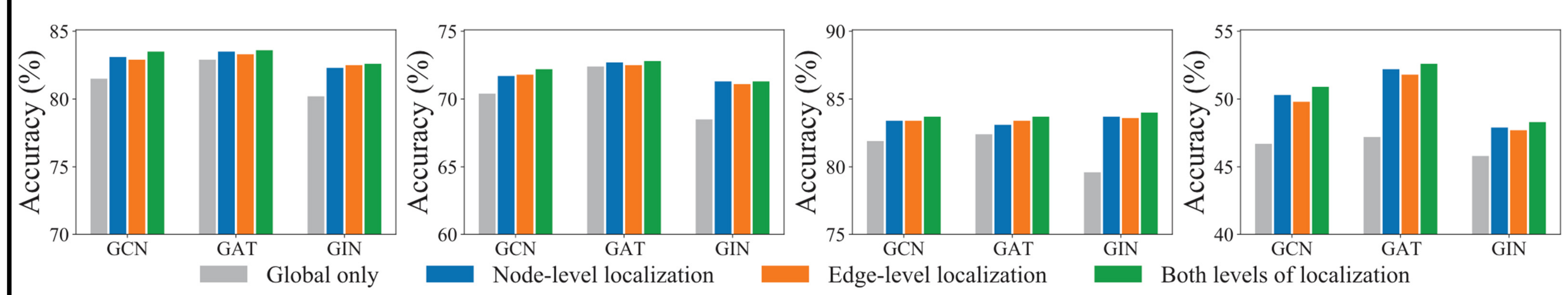
- Evaluation
  - Accuracy, Micro-F
- Baselines
  - Embedding models: DeepWalk [1], Planetoid [2]
  - GNN models: GCN [3], GAT [4], GIN [5]
  - GNN-FiLM [6]: GCN-FiLM, GAT-FiLM, GIN-FiLM

Node classification

Methods	# Params (Cora)	Cora		Citeseer		Amazon		Chameleon	
		Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F	Accuracy	Micro-F
DeepWalk	693K	73.8±0.3	74.9±0.1	61.6±0.2	60.5±1.0	80.1±1.6	77.3±1.3	41.2±1.3	40.1±1.1
Planetoid	345K	66.1±0.4	64.5±0.5	64.5±0.3	62.9±0.4	69.8±1.7	64.5±1.5	39.3±1.8	37.7±1.7
GCN	11K	81.5±0.7	80.8±0.5	70.4±0.5	68.3±0.7	81.9±0.5	81.0±0.8	46.7±4.3	46.4±2.4
GCN-64	92K	82.0±0.3	80.9±0.3	71.1±0.3	69.2±0.4	82.1±0.5	81.2±0.8	48.3±3.3	46.3±1.8
GCN-96	138K	81.9±0.2	80.8±0.3	71.3±0.4	69.4±0.5	82.2±0.4	81.5±0.7	45.5±2.4	43.8±2.5
GCN-FiLM	35K	78.1±0.6	76.9±0.5	69.8±1.1	67.9±1.0	79.2±1.0	77.1±1.5	42.8±1.1	39.9±1.3
LGCN	104K	**83.5**±0.3	**82.1**±0.4	**72.2**±0.4	**70.2**±0.4	**83.7**±1.5	**82.3**±2.0	**50.9**±1.1	**49.7**±0.7
(improv.)	-	(1.8%)	(1.5%)	(1.3%)	(1.2%)	(1.8%)	(1.0%)	(5.4%)	(7.1%)
GAT	92K	82.9±0.6	82.0±0.6	72.4±0.7	70.4±0.8	82.4±1.3	80.1±1.9	47.2±1.1	46.2±2.1
GAT-64	738K	83.1±0.4	81.9±0.6	71.6±1.5	69.8±1.6	83.0±0.9	81.2±1.4	51.2±1.5	50.2±1.3
GAT-96	1108K	83.2±0.6	81.9±0.6	71.4±0.9	69.6±0.9	83.1±1.0	81.5±1.4	51.9±1.2	50.2±1.8
GAT-FiLM	277K	82.0±0.5	80.6±0.6	71.2±1.0	69.2±1.1	83.3±0.6	81.9±0.8	46.8±5.7	45.1±5.2
LGAT	836K	**83.6**±0.4	**82.3**±0.4	**72.8**±0.4	**70.8**±0.5	**83.7**±0.7	**82.3**±0.8	**52.6**±1.0	**51.1**±0.9
(improv.)	-	(0.5%)	(0.4%)	(0.6%)	(0.6%)	(0.5%)	(0.5%)	(1.3%)	(1.8%)
GIN	11K	80.2±0.5	78.8±0.3	68.5±0.7	66.5±1.0	79.6±1.7	78.5±2.6	45.8±3.0	41.2±4.0
GIN-64	92K	80.3±1.1	79.1±1.0	67.8±1.5	66.1±1.1	79.8±1.1	79.0±1.4	45.7±4.5	40.7±5.7
GIN-96	138K	79.9±1.1	78.9±1.0	68.6±1.4	66.6±1.6	80.2±2.1	79.0±3.2	45.9±3.5	41.5±4.1
GIN-FiLM	35K	79.8±0.7	78.5±0.5	67.7±1.4	65.8±1.5	78.6±2.8	77.2±3.3	38.8±2.6	34.2±2.9
LGIN	126K	**82.6**±0.8	**81.6**±0.8	**71.3**±0.4	**69.5**±0.5	**84.0**±1.2	**82.7**±1.7	**48.3**±1.9	**47.3**±1.9
(improv.)	-	(2.9%)	(3.2%)	(3.9%)	(4.4%)	(4.7%)	(4.7%)	(5.2%)	(14.0%)

- LGNN consistently achieves significant performance boosts
- GAT-based models generally attain better performance than GCN- and GIN-based models
- Increasing the number of parameters alone cannot achieve the effect of localization

Ablation study



(a) Cora (b) Citeseer (c) Amazon (d) Chameleon

- Utilizing only one module consistently outperforms the global model
- The node-level localization tends to perform better than edge-level localization.
- Modeling both jointly results in the best performance

Conclusions

Motivation
- We identified the need to **localize GNNs** for different nodes

Proposed model: LGNN
- Encode graph-level general patterns using a **global weight matrix**
- **Node-level** and **edge-level** localization

Experiments
- Extensive experiments demonstrate that LGNN significantly outperforms state-of-the-art GNNs.

Reference

[1] Perozzi B, et al. 2014. Deepwalk: Online learning of social representations. KDD.
[2] Yang Z, et al. 2016. Revisiting semi-supervised learning with graph embeddings. ICML.
[3] Kipf, T. N., et al. 2017. Semi-supervised classification with graph convolutional networks. ICLR.
[4] Veličković, P., et al. 2018. Graph attention networks. ICLR.
[5] Xu K, et al. 2019. How powerful are graph neural networks? ICLR.
[6] Brockschmidt M. 2020. Gnn-film: Graph neural networks with feature-wise linear modulation. ICML.