UNIVERSITÄT BIELEFELD
Faculty of Technology

Graph Learning by Dynamic Sampling

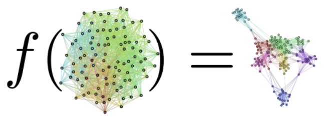
Luca Hermes[1], Aleksei Liuliakov[1], Malte Schilling[2]
[1]Machine Learning Group, Bielefeld University; [2]Autonomous Intelligent Systems Group, University of Münster

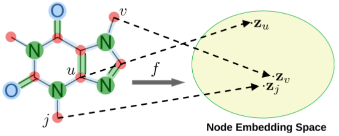
CITEC
dataninja.nrw

Intro: Graph Neural Networks

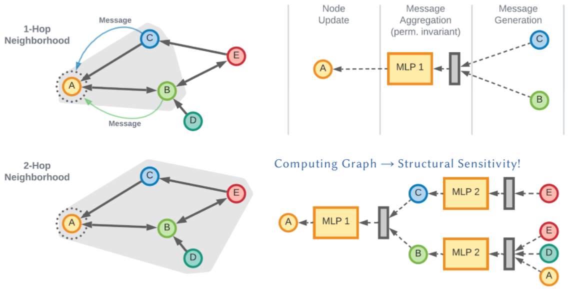
General GNN Framework

- Functions that embed nodes based on structure and node features
- Two nodes in a similar structural context should be mapped to similar locations in the embedding space



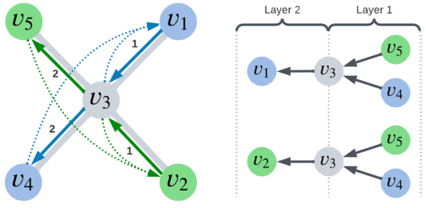


Message Passing Neural Networks [1, 2]



Drawbacks of MPNNs

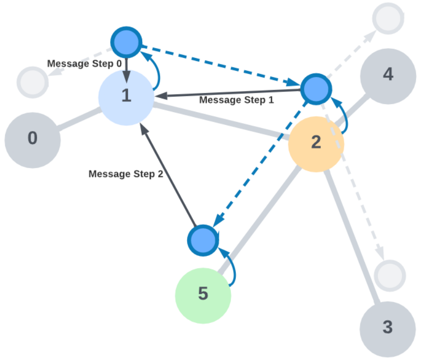
- Feature exchange between nodes *4* and *1* requires passing across node *3*
- Bridging longer distances on the graph leads to problems like *over-smoothing* [3]
- Worse at information bottlenecks
- Can prevent learning in heterophilic settings (predominantly dissimilar nodes are connected) [4]
- Here we present an approach to mitigate bottlenecks and over-smoothing

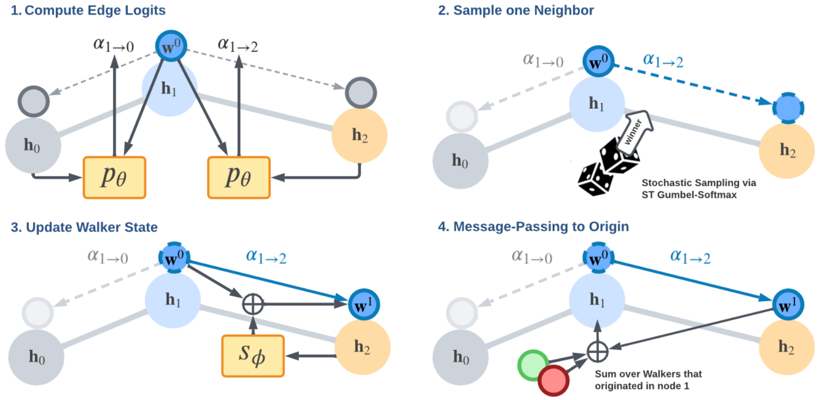


2-Step Dynamic Sampling

2-Layer Graph Convolution

Dynamic Sampling Graph Neural Network (DSGNN)





1. Compute Edge Logits between nodes u and v

$$p_\theta(\mathbf{u},\mathbf{v}) = \mathbf{a}^T\left(\text{MLP}_1(\mathbf{u})|\text{MLP}_2(\mathbf{v})\right)$$

2. Sample one Neighbor from Neighbor Set X

$$\bar{p}_b(X) = \text{softmax}_\tau(X+G)$$

$$\bar{p}_f(X) = \text{argmax}_{\forall x\in X}(\text{softmax}_\tau(X+G))$$

3. Update Walker State with the sampled Node

$$\mathbf{w}_i^{t+1} = s_\phi(\mathbf{w}_i^t, \mathbf{x}_j^t) = \text{MLP}_\phi(\mathbf{x}_j^t) + \mathbf{w}_i^t$$

Results and Conclusion

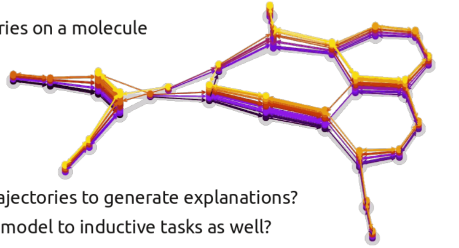
Table 1: Results for transductive graph benchmarks. The values are accuracies averaged over all 10 data splits and we include the standard deviation. The best performing model is highlighted in bold and the second best is marked with italic.

	Texas	**Wisconsin**	**Actor**	**Squirrel**	**Chameleon**	**Cornell**	**Citeseer**	**Pubmed**	**Cora**
homophily	0.11	0.20	0.22	0.22	0.24	0.13	0.74	0.80	0.81
# Nodes	183	251	7600	5201	2277	183	3327	19717	2708
# Edges	325	515	33391	217073	36101	298	4614	44325	5278
# Classes	5	5	5	5	5	5	6	3	7
DSGNN-DP	**86.25**±3.32	84.27±3.12	*37.62*±0.60	49.50±0.98	*68.07*±1.54	74.58±4.56	76.57±0.85	88.38±0.41	85.87±0.65
DSGNN-GAT	85.42±3.93	83.66±2.50	37.43±0.82	47.87±1.30	63.77±1.31	72.19±3.37	76.25±0.86	88.32±0.80	85.80±0.87
$O(d)$-NSD	*85.95*±5.51	**89.41**±4.74	**37.81**±1.15	**56.34**±1.32	68.04±1.58	84.86±4.71	76.70±1.57	89.49±0.40	86.90±1.13
GGCN	84.86±4.55	86.86±3.29	37.54±1.56	*55.17*±1.58	**71.14**±1.84	**85.68**±6.63	77.11±1.45	89.15±0.37	*87.95*±1.05
H2GCN	84.86±7.23	*87.65*±4.98	35.70±1.00	36.48±1.86	60.11±2.15	82.70±5.28	77.11±1.57	89.49±0.38	87.87±1.20
GCNII	77.57±3.83	80.39±3.40	37.44±1.30	38.47±1.58	63.86±3.04	77.86±3.79	*77.33*±1.48	**90.15**±0.43	**88.37**±1.25
Geom-GCN	66.76±2.72	64.51±3.66	31.59±1.15	38.15±0.92	60.00±2.81	60.54±3.67	**78.02**±1.15	*89.95*±0.47	85.35±1.57
GCN	55.14±5.16	51.76±3.06	27.32±1.10	53.43±2.01	64.82±2.24	60.54±5.30	76.50±1.36	88.42±0.50	86.98±1.27
GAT	52.16±6.63	49.41±4.09	27.44±0.89	40.72±1.55	60.26±2.50	61.89±5.05	76.55±1.23	87.30±1.10	86.33±0.48
MLP	80.81±4.75	85.29±3.31	36.53±0.70	28.77±1.56	46.21±2.99	81.89±6.40	74.02±1.90	87.16±0.37	75.69±2.00

- DSGNN shows some benefits in performance, although not consistently
- Seems to work best in heterophilic settings
- Two types of edge model (GAT-style [2] and dot product attention) both work well, seems to depend on dataset

Qualitative Results

- Sampling trajectories on a molecule



Future Work

- Can we use the trajectories to generate explanations?
- Can we apply this model to inductive tasks as well?
- Our current state model is very rudimentary and can be improved upon

References
[1] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations, 2017.
[2] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Li`o, and Y. Bengio. Graph attention networks, 2017.
[3] Q. Li, Z. Han, and X. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. CoRR, abs/1801.07606, 2018.
[4] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, and D. Koutra. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. arXiv e-prints, page arXiv:2102.06462, Feb. 2021.