Supplemental Material for Stochastic Blockmodels meet Graph Neural Networks

1. Hyparameter Settings

QUANTITATIVE RESULTS:

The framework proposed uses a stick-breaking IBP prior which has two parameters: α and K. The parameter α is the initial guess of the number of non-zero entries in the binary vector b_n and K is the truncation parameter. In the experiments, $\alpha \in \{5, 10, 20, 50, 100\}$. In general, a higher value of the α parameter worked better for DGLFRM-B and LFRM, as compared to the α value in the DGLFRM model. This difference in α reflects the inherent capacity of the latent space of these models. The embedding learned by DGLFRM, while being highly sparse, are in real space resulting in more capacity to represent data as compared to the binary latent space in DGLFRM-B and LFRM.

The encoder network for DGLFRM and DGLFRM-B had two non-linear GCN layers. The length of the first non-linear layer was fixed to 32/64 for the datasets which had side-information (Cora, Citeseer and Pubmed), and otherwise was set to 128/256. The second layer of the GCN encoder had $K \in \{50, 100, 200\}$ hidden units. The decoder network for DGLFRM and DGLFRM-B had two layers with dimension 32 and 16. All the models were trained for 500-1000 iterations using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.01. We used 0.5 dropout. The temperature parameter of the Binary Concrete distribution (Maddison et al., 2017) was 0.5 for the prior and 1.0 for the posterior.

QUALITATIVE RESULTS:

For experiments on the synthetic data (with 100 nodes and 10 communities), the DGLFRM model had two GCN encoding layers with 32 and $K = 10$ hidden units, and the decoder had a simple inner-product layer. The VGAE model had the same set of hyperparameters as above. The qualitative experiment on the NIPS12 co-authorship dataset had two hidden layers with 64 and $K = 10$ hidden units. The α parameter for this experiment was fixed to 2.

2. K-means on VGAE embeddings

Variational Graph Autoencoder (VGAE), unlike the proposed model, is not able to learn embeddings which are readily interpretable. It requires additional processing such as K-Means over the learned embeddings for node clustering. Moreover, a method like K-Means does not result in overlapping communities. In this section, we compare the clusters obtained after applying K-means on embeddings learned from VGAE with the readily available overlapping communities obtained from our framework.

We use K-means to find clusters for NIPS12 (3134 authors) co-authorship data on the node embeddings learned using VGAE. The K-means results are shown in Table 1. We performed two experiments with different k-means cluster hyperparameter K (K=5 and K=20). We also show the clusters (communities) which our model was readily able to infer for reference Table 2. We only show prominent authors and their clusters for both the models.

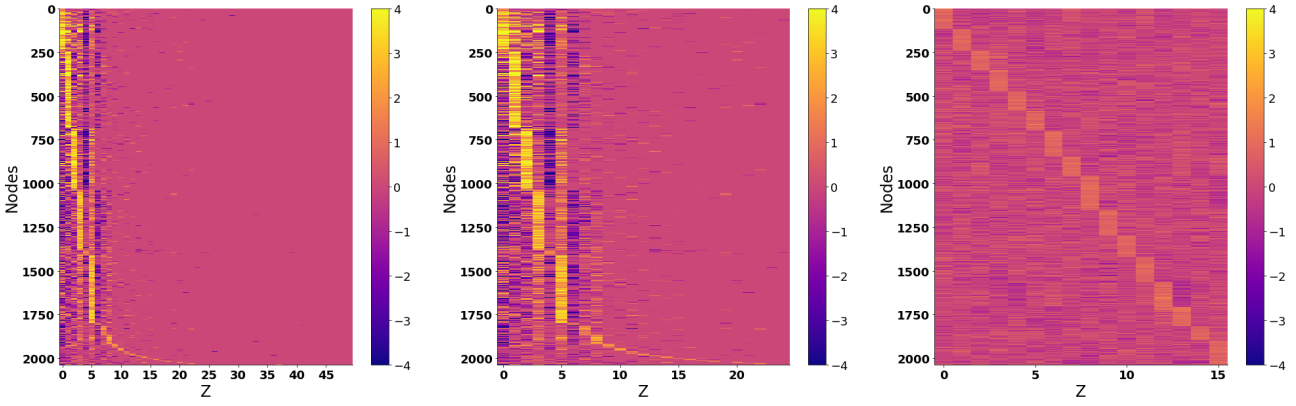
As we see in Table 1, ad-hoc post-processing of embeddings may break some relevant coherent communities which were inferred by our model. It is also important to note that, unlike our model, k-means has no strength indicator for community membership.

3. Latent Structure on NIPS12

The latent structure of the NIPS12 dataset learned by DGLFRM and VGAE is shown in Figure 1. In this experiment, the truncation parameter for the stick-breaking prior is 50. As shown in Figure 1(b), the posterior inference in DGLFRM is naturally able to "turn off" the unnecessary columns in $\mathbf{Z}$. The average number of active communities for each node was found to be 8. The sparse nature of the embedding matrix allows us to consider each column as a possible community of a given node. For visualization, we have ordered the indices of the communities (columns of $\mathbf{Z}$) such that the community with higher active nodes has a lower index in the visualization. For the VGAE model, we used a two layer GCN with dimensions 32 and 16. Figure 1(c) depicts the dense node embedding learned by VGAE.

3.1. Effect of Side Information

We also perform an experiment to investigate the model's ability to leverage side information associated with nodes. For this experiment, we ran our model on three datasets (Cora, Citeseer and Pubmed) with and without node features.



(a) DGLFRM (b) DGLFRM (Filtered) (c) VGAE

Figure 1. (a-c) The latent structure on NIPS12 dataset using DGLFRM and VGAE. Both the models had the same encoder and latent dimension. The latent structure learned by DGLFRM was filtered by removing the columns which were inactive for all nodes. DGLFRM can effectively infer the "active" communities.

Table 1. Example of NIPS12 communities inferred by k-means clustering (post-processing step) on the embeddings learned using VGAE. Authors have hard-assignments (memberships) in these communities.

Cluster (K=5)	Authors
Cluster 1	Hinton G, Dayan P, Jordan M, Tang A, Sejnowski T, Willams C
Cluster 2	Weinshall D, Rinott Y, Barto A, Singh S, Sutton R, Giles C, Connolly C, Baldi P, Precup D
Cluster 3	Thrun S, Shibata T, Stein C, Peper F, Michel A, Druzinsky R, Abu-Mostafa Y
Cluster 4	LeCun Y, Pearlmutter B
Cluster (K=20)	**Authors**
Cluster 1	Hinton G, Williams C
Cluster 2	Jordan M, Connolly C, Barto A, Singh S, Sutton R
Cluster 3	Michel A, Tang A
Cluster 4	Dayan P, Sejnowski T
Cluster 5	Thrun S, Peper F
Cluster 6	Baldi P, Weinshall D
Cluster 7	Shibata T, Druzinsky R
Cluster 8	Stein C
Cluster 9	Precup D
Cluster 10	Giles C
Cluster 11	Pearlmutter B
Cluster 12	LeCun Y
Cluster 13	Rinott Y
Cluster 14	Abu-Mostafa Y

Table 2. Example of communities inferred by our model on NIPS data. Authors ordered by strength of membership in these communities.

Cluster	Authors
Probabilistic Modeling	**Sejnowski T**, Hinton G, Dayan P, Jordan M, Williams C
Reinforcement Learning	Barto A, Singh S, Sutton R, Connolly C, Precup D
Robotics/Vision	Shibata T, Peper F, Thrun S, Giles C, Michel A
Computational Neuroscience	Baldi P, Stein C, Rinott Y, Weinshall D, Druzinsky R
Neural Networks	Pearlmutter B Abu-Mostafa Y, LeCun Y, **Sejnowski T**, Tang A

We compare the AUC-ROC results in Fig. 2. As expected, when using the node side information, the model performs better as compared to the case when it ignores the side information.

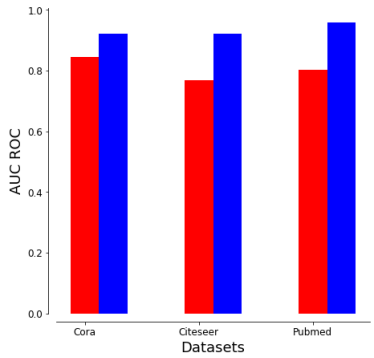


Figure 2. Red: Without side information. Blue: With side information.

References

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.