Learning interpretable hierarchical node representations via Ladder Gamma VAE

Arindam Sarkar[*†]
Amazon

Nikhil Mehta[*†]
Duke University

Piyush Rai
IIT Kanpur

Abstract

We present a probabilistic framework for community discovery and link prediction for graph-structured data, based on a novel, gamma ladder variational autoencoder (VAE) architecture. We model each node in the graph via a deep hierarchy of gamma-distributed embeddings and define each link probability via a nonlinear function of the bottom-most layer's embeddings. The gamma latent variables naturally result in non-negativity and sparsity of the learned embeddings, and facilitate direct interpretation as membership of nodes into (possibly multiple) communities/topics. The proposed ladder-based recognition model allows fast inference over the hierarchical embeddings. We compare our model with other state-of-the-art methods and report encouraging quantitative and qualitative results.

1 Introduction

Representation learning for the nodes in a graph is an important problem in a wide range of applications involving graph-structured data, such as community discovery, link-prediction, node classification, etc [4]. Some of the prominent prior works in this direction include stochastic blockmodels and variants [13, 12, 1, 10] and, more recently, graph neural networks [14, 8, 3]. While stochastic blockmodels and their variants are effective at learning the underlying latent structure (e.g., community structure) of the graph using latent variables that denote node membership to communities, the graph neural network based methods, such as graph convolutional networks (GCN) [8] and its variants [3] are appealing since they enable learning *multilayer* representation for the nodes in the graph, which has been shown to achieve impressive results on link-prediction and node classification.

Despite providing nice interpretability for the node embeddings, the powerful variants of stochastic blockmodels such as mixed-membership blockmodels [1] and overlapping stochastic blockmodels [12, 19] are especially difficult to do inference on (relying on expensive MCMC or variational inference), and are difficult to scale. On the other hand, the recently proposed graph neural networks lack a proper generative story, do not have a mechanism to do model selection (e.g., inferring the size of node embeddings), and the learned embeddings do not have direct interpretability (required for tasks such as community discovery).

In this work, we develop a deep generative framework for graph-structured data that enjoys the natural advantages of stochastic blockmodels and graph neural networks, while also addressing their limitations/shortcomings in a principled manner. Our framework is based on a novel, *gamma* ladder variational autoencoder (VAE) architecture, which allows each node in the graph to be modeled by *multiple* layers of gamma-distributed latent variables (which represent multiple layers of embeddings for each node). The probability of each link in the graph is a nonlinear function (modeled by a deep neural network) of the node embeddings of the associated nodes. While existing ladder VAE architectures [15] typically assume dense, Gaussian distributed latent variables in each layer, the

[*] Equal Contribution. Correspondence to <arindamsarkar93@gmail.com>

[†] Majority of the work was done when Arindam and Nikhil were at IIT Kanpur.

Graph Representation Learning Workshop (NeurIPS 2019), Vancouver, Canada.

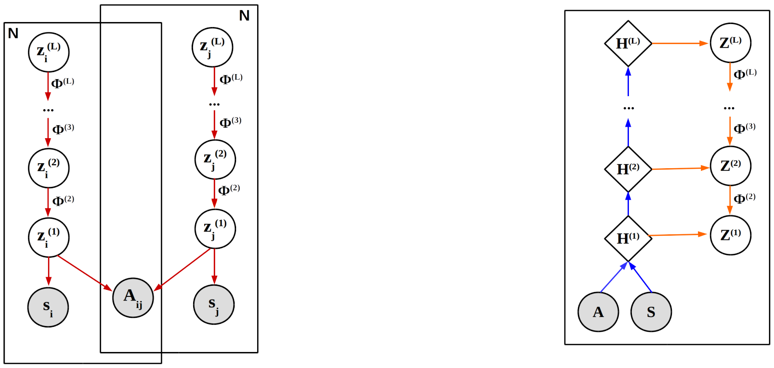


Figure 1: (Left) The decoder/generator network depicting link (A_{ij}) generation using the node embeddings. (Right) The inference/recognition network which takes the adjacency matrix $\mathbf{A}$ and the side information $\mathbf{S}$ as input. For the upward deterministic pass (blue), a graph encoder (GCN) is used. The downward pass (orange) is the deep hierarchy of latent variables. The model uses information sharing scheme between the inference and generator network (left to right). While doing inference, the intermediate layers in the upward pass are conditioned on the complete adjacency matrix $\mathbf{A}$. Here $\mathbf{H}^{(l)} = \left[\boldsymbol{h}_1^{(l)}, ..., \boldsymbol{h}_N^{(l)}\right]^T$ and $\mathbf{Z}^{(l)} = \left[\boldsymbol{z}_1^{(l)}, ..., \boldsymbol{z}_N^{(l)}\right]^T$.

gamma-distributed embeddings in our ladder VAE architecture naturally provide sparsity [2, 19] and interpretability of the node embeddings.

2 Preliminaries

We will introduce the notation as we go. We will first briefly describe the Graph Convolution Network (GCN) [8] and the Ladder Variational Autoencoder (VAE) [15]. GCN uses a series of convolutional operators to find a nonlinear deterministic embedding for each node in a graph. Given an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, the forward propagation rule for each layer l in GCN is defined as $\mathbf{H}^{(l)} = g(\mathbf{A}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)})$ where $\mathbf{H}^{(0)} = \mathbf{S}$ ($\mathbf{S} = \mathbf{I}$ when no side information is present), $\mathbf{W}^{(l)}$ is the weight matrix for the layer l and $g(.)$ is the element-wise non-linearity. GCN have recently emerged as a flexible encoder for graphs (similar in spirit to CNNs for images) which makes them an ideal choice for the encoding step in our VAE based generative framework. As we show in § 5, the vanilla GCN model can not learn interpretable node embeddings and hence cannot be used for community detection for nodes unlike our model.

The ladder VAE [15] is an improvement over the standard VAE [7] by having multiple stochastic latent variables in VAE based deep generative models (note that the standard VAE has only a single layer of stochastic latent variables and multiple layers of deterministic variables). Ladder VAE does inference over multiple stochastic variables via an information sharing scheme between the upward deterministic pass and the downward stochastic pass during the inference of latent variables. In particular, each stochastic latent variable is conditioned on a deterministic variable from the upward pass and a stochastic latent variable from the downward pass. In § 4, we leverage this information sharing scheme to do inference over a hierarchy of interpretable node embeddings.

3 Gamma Ladder VAE for Graphs

The generative process for our gamma ladder VAE for modeling graph-structured data is shown in Fig. 1 (Left). We assume that an observed edge A_{ij} between nodes i and j is associated with a deep hierarchy of latent variables $\{\boldsymbol{z}_i^{(\ell)}\}_{\ell=1}^L$ and $\{\boldsymbol{z}_j^{(\ell)}\}_{\ell=1}^L$. The latent variables $\{\boldsymbol{z}_i^{(\ell)}\}_{\ell=1}^L$ denote the embeddings of the node i, where $\boldsymbol{z}_i^{(\ell)} \in \mathbb{R}_+^{K_\ell}$ and K_ℓ is the embedding size in layer ℓ. The deep hierarchy of gamma-distributed embeddings is generated as follows

$$\boldsymbol{z}_i^{(L)} \sim \text{Gam}(\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{r}}^{(L)}), \;\; \ldots \;\; \boldsymbol{z}_i^{(\ell)} \sim \text{Gam}(\boldsymbol{\Phi}^{(\ell+1)}\boldsymbol{z}_i^{(\ell+1)}, \hat{\boldsymbol{r}}^{(\ell)}), \;\; \ldots, \;\; \boldsymbol{z}_i^{(1)} \sim \text{Gam}(\boldsymbol{\Phi}^{(2)}\boldsymbol{z}_i^{(2)}, \hat{\boldsymbol{r}}^{(1)}) \quad (1)$$

Here $\boldsymbol{\Phi}^{(\ell)} = [\boldsymbol{\phi}_1^{(\ell)}, \boldsymbol{\phi}_2^{(\ell)}, ...\boldsymbol{\phi}_{K_\ell}^{(\ell)}]$ denotes a non-negative $K_{\ell-1} \times K_\ell$ transformation matrix, with each column $\boldsymbol{\phi}_{k_\ell}^{(\ell)} \in \mathbb{R}_+^{K_{\ell-1}}$ summing to one. An especially appealing aspect of the above hierarchical construction is that the non-negativity and sparsity of the gamma latent variables allows direct interpretability of the node embeddings as *communities* [19]. In particular, each component of the node embedding vector $\boldsymbol{z}_i^{(\ell)} \in \mathbb{R}_+^{K_\ell}$ denotes the strength of membership of node i into one of the K_ℓ communities. Moreover, since our model learns a multilayer embedding for each node, it can infer communities at multiple layers of granularities [5].

We assume that each link A_{ij} is generated as $A_{ij} \sim \text{Bern}(p_{ij})$; where $p_{ij} = f(\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(1)})$. Here $f(.,.)$ can be any differentiable function which takes input two vectors to give a probability score.

4 Inference

Exact inference in our model is intractable, partly because of the multiple layers of embeddings and partly because of the choice of Gamma distribution as prior. Thus we use stochastic gradient variational Bayes (SGVB) [6, 7] to perform inference for our model. Figure 1 (Right) shows our inference network used for our recognition model. We approximate the model's true posterior $p(\{\boldsymbol{z}_i^{(\ell)}\}_{i=1,\ell=1}^{N,L} | \mathbf{A}, \mathbf{S})$ with a variational posterior $q(\{\boldsymbol{z}_i^{(\ell)}\}_{i=1,\ell=1}^{N,L})$. Unlike traditional ladder VAE, our framework uses gamma latent variables instead of Gaussian latent variables. While gamma distribution would have been a more suitable variational distribution in our case, gamma random variables do not have an easy reparameterization, which makes it difficult to apply SGVB. To address this issue, we approximate the variational posterior of the node embeddings using Weibull distributions. Weibull and gamma distribution, both being special cases of the generalized gamma distribution [16], are closely related and have similar PDFs. Weibull can easily be reparameterized [18] as follows

$$q(\boldsymbol{z}_i^{(\ell)}) = \text{Weibull}(\boldsymbol{\alpha}^{(\ell)}, \boldsymbol{\beta}^{(\ell)}), \quad U \sim \text{Uniform}(0,1), \quad \boldsymbol{z}_i^{(\ell)} = \boldsymbol{\beta}^{(\ell)}(-\ln(1-U))^{\frac{1}{\boldsymbol{\alpha}^{(\ell)}}} \tag{2}$$

Our inference network consists of a nonlinear graph encoder to learn the parameters of the variational posterior as defined in Equation 2. The encoder model is based on the recognition network used in ladder VAE [15] (but with Weibull approximate latent variables) that first uses a deterministic upward pass to compute the approximate likelihood contributions from the data

$$\{\mathbf{H}^{(\ell)}\}_{\ell=1}^{L} = \text{GCN}(\mathbf{A}, \mathbf{S}), \quad \hat{\mathbf{R}}^{(\ell)} = \text{Softplus}(\mathbf{H}^{(\ell)}\mathbf{W}_r^{(\ell)}), \quad \hat{\mathbf{S}}^{(\ell)} = \text{Softplus}(\mathbf{H}^{(\ell)}\mathbf{W}_s^{(\ell)}) \tag{3}$$

The upward pass is followed by a stochastic downward pass to compute the generative distributions and their variational parameters. We assume mean-field approximation and factorize the variational distributional as $\prod_{i=1}^{N} q(\boldsymbol{z}_i^{(L)} | \boldsymbol{h}_i^{(L)}) \prod_{\ell=1}^{L-1} q(\boldsymbol{z}_i^{(\ell)} | \boldsymbol{h}_i^{(\ell)}, \boldsymbol{z}_i^{(\ell+1)})$. The downward pass can be done recursively as

$$q(\boldsymbol{z}_i^{(L)} | \boldsymbol{h}_i^{(L)}) = \text{Weibull}(\boldsymbol{\alpha}_i^{L}, \boldsymbol{\beta}_i^{L}), \quad q(\boldsymbol{z}_i^{(\ell)} | \boldsymbol{h}_i^{(\ell)}, \boldsymbol{z}_i^{(\ell+1)}) = \text{Weibull}(\boldsymbol{\alpha}_i^{\ell}, \boldsymbol{\beta}_i^{\ell}) \tag{4}$$

where $\boldsymbol{\alpha}_i^{\ell} = \hat{\boldsymbol{s}}_i{}^{(L)}$, $\boldsymbol{\beta}_i^{\ell} = \hat{\boldsymbol{r}}_i{}^{(L)}$ if $\ell = L$, otherwise $\boldsymbol{\alpha}_i^{\ell} = \hat{\boldsymbol{s}}_i{}^{(\ell)} + \boldsymbol{\Phi}^{\ell+1}\boldsymbol{z}_i^{\ell+1}, \quad \boldsymbol{\beta}_i^{\ell} = \hat{\boldsymbol{r}}_i^{(\ell)}$. Following the SGVB training scheme [6], we train our gamma ladder VAE by maximizing the evidence lower bound (ELBO) where all $q(\boldsymbol{z}_n^{(\ell)})$, except for the top-layer $q(z_n)^{(L)}$, are conditioned on $\boldsymbol{z}_n^{(\ell+1)}$.

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{E}[\ln p(A_{ij} | \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(1)})] - \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\Big[\ln q(\boldsymbol{z}_n^{(l)}) - \ln p(\boldsymbol{z}_n^{(l)})\Big] \tag{5}$$

5 Experiments

We refer to our model as **LGVG** (**L**adder **G**amma **V**ariational Autoencoder for **G**raphs) and its version with side-information as **LGVG-X**. The LGVG-X variant generates the side-information (when available) using the bottom-most node embedding. We first evaluate our model on the link-prediction task over 4 real-world benchmark graph datasets (see table 1) comparing it against other methods. As shown in Table 1, our model has competitive performance with other state-of-the-art methods.

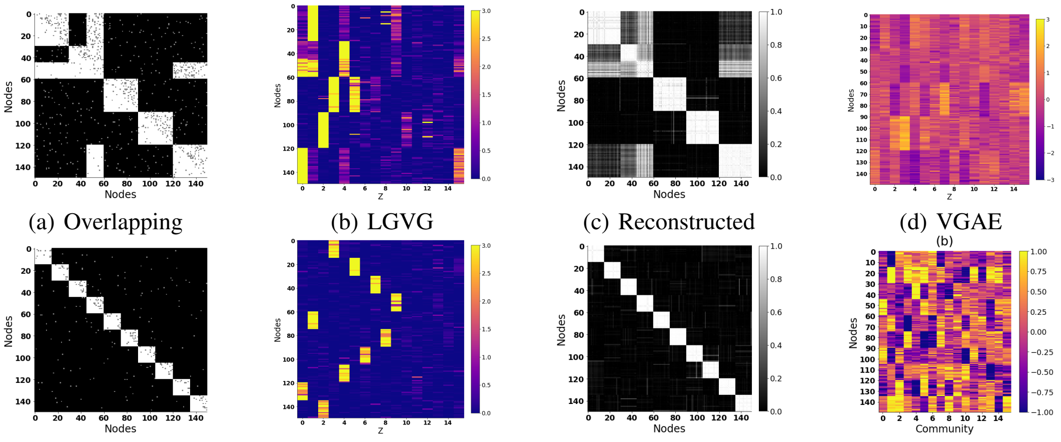
Next, we evaluate our model on synthetic datasets for qualitative analyis (e.g., inferred node embeddings/communities). We generate two synthetic datasets having 150 nodes. There are 10 overlapping communities in the first dataset and 10 non-overlapping in the second. Figure 5 shows the learned latent embeddings by LGVG in comparison to the embedding learned using the vanilla-GCN.

We also conduct a qualitative experiment on the real-world NIPS12 co-authorship dataset and examine the community structure discovered by our model. See table 3 in Appendix which shows communities learned by LGVG. Our model can learn a hierarchical community structure, and we show an illustration for the same in Fig. 4.

6 Conclusion

Table 1: Average Precision (AP).

Method	NIPS12	Cora	Citeseer	Pubmed
SC [17]	0.9022 ± 0.0003	0.8850 ± 0.0000	0.8500 ± 0.0001	0.8780 ± 0.0001
DW [14]	0.8634 ± 0.0000	0.8500 ± 0.0000	0.8360 ± 0.0001	0.8410 ± 0.0000
VGAE [9]	0.9111 ± 0.0025	0.9260 ± 0.0001	0.9200 ± 0.0002	0.9470 ± 0.0002
DGLFRM [11]	0.9005 ± 0.0027	0.9376 ± 0.0022	0.9438 ± 0.0073	0.9497 ± 0.0035
EPM-SGVB [19]	0.9086 ± 0.0129	0.8666 ± 0.0109	0.8259 ± 0.0172	0.8600 ± 0.0047
LGVG	0.9260 ± 0.0068	0.9254 ± 0.0068	0.9130 ± 0.0112	0.9545 ± 0.0024
LGVG-X	**0.9260** ± 0.0068	**0.9502** ± 0.0061	**0.9624** ± 0.0067	**0.9559** ± 0.0017



(a) Overlapping (b) LGVG (c) Reconstructed (d) VGAE

(e) Nonoverlapping (f) LGVG (g) Reconstructed (h) VGAE

Figure 2: (a-d) Experiments on the overlapping synthetic dataset. (a) Training adjacency matrix (Black, White and Gray denote no-link, link and the 15% masked respectively). (b) Last layer node embeddings learned by the LGVG. (c) Reconstructed graph using LGVG model indicating probability of link between nodes (white (black) suggest high probability of link (no-link)). (d) Node embeddings learned by the VGAE model. (e-h) Experiments on the non-overlapping synthetic dataset. (e) The training adjacency matrix. (f) The node embeddings learned by the LGVG. (g) Reconstructed graph using LGVG model. (h) Node embeddings learned by the VGAE model.

We have presented a novel, gamma ladder VAE model for graph-structured data, that enables us to infer multilayered embeddings (in form of multiple layers of stochastic variables) for the nodes in a graph. Besides having strong predictive power, the embeddings learned by our model are sparse and directly interpretable. Our model outperforms recently proposed deep generative models that are based on a vanilla VAE architecture, both on quantitative metrics as well as on qualitative analysis tasks (being able to learn embeddings which are directly interpretable, eliminating the need for an extra clustering/dimensionality reduction step which is required for dense Gaussian embeddings) using the learned deep representations of the nodes. We believe our model to be an important first step in bringing together the interpretability of hierarchical, multilayer latent variable models such as ladder variational autoencoders and the strong representational power of graph encoders, such as graph convolutional networks, for modeling graph-structured data.

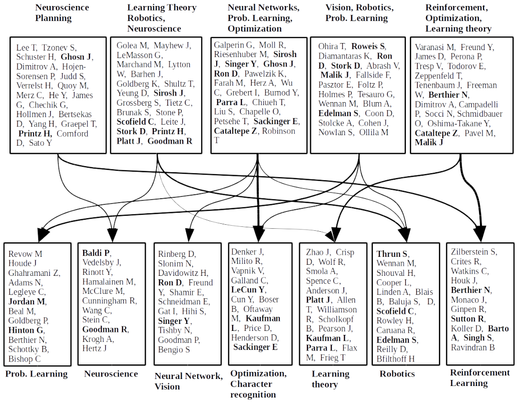

Neuroscience Planning
Learning Theory Robotics, Neuroscience
Neural Networks, Prob. Learning, Optimization
Vision, Robotics, Prob. Learning
Reinforcement, Optimization, Learning theory
Lee T, Tzonev S, Schuster H, Ghosn J, Dimitrov A, Hojen-Sorensen P, Judd S, Verrelst H, Quoy M, Merz C, He Y, James G, Chechik G, Hollmen J, Bertsekas D, Yang H, Graepel T, Printz H, Cornford D, Sato Y
Golea M, Mayhew J, LeMasson G, Marchand M, Lytton W, Barhen J, Goldberg K, Shultz T, Yeung D, Sirosh J, Grossberg S, Tietz C, Brunak S, Stone P, Scofield C, Leite J, Stork D, Printz H, Platt J, Goodman R
Galperin G, Moll R, Riesenhuber M, Sirosh J, Singer Y, Ghosn J, Ron D, Pawelzik K, Farah M, Herz A, Wu C, Grebert I, Burnod Y, Parra L, Chiueh T, Liu S, Chapelle O, Petsche T, Sackinger E, Cataltepe Z, Robinson T
Ohira T, Roweis S, Diamantaras K, Ron D, Stork D, Abrash V, Malik J, Fallside F, Pasztor E, Foltz P, Holmes P, Tesauro G, Wennan M, Blum A, Edelman S, Coon D, Stolcke A, Cohen J, Nowlan S, Ollila M
Varanasi M, Freund Y, James D, Perona P, Tresp V, Todorov E, Zeppenfeld T, Tenenbaum J, Freeman W, Berthier N, Dimitrov A, Campadelli P, Socci N, Schmidbauer O, Oshima-Takane Y, Cataltepe Z, Pavel M, Malik J
Revow M Houde J Ghahramani Z, Adams N, Legleye C, Jordan M, Beal M, Goldberg P, Hinton G, Berthier N, Schottky B, Bishop C
Baldi P, Vedelsby J, Rinott Y, Hamalainen M, McClure M, Cunningham R, Wang C, Stein C, Goodman R, Krogh A, Hertz J
Rinberg D, Slonim N, Davidowitz H, Ron D, Freund Y, Shamir E, Schneidman E, Gat I, Hihi S, Singer Y, Tishby N, Goodman P, Bengio S
Denker J, Milito R, Vapnik V, Galland C, LeCun Y, Cun Y, Boser B, Oftaway M, Kaufman L, Price D, Henderson D, Sackinger E
Zhao J, Crisp D, Wolf R, Smola A, Spence C, Anderson J, Platt J, Allen T, Williamson R, Scholkopf B, Pearson J, Kaufman L, Parra L, Flax M, Frieg T
Thrun S, Wennan M, Shouval H, Cooper L, Linden A, Blais B, Baluja S, D, Scofield C, Rowley H, Caruana R, Edelman S, Reilly D, Bfilthoff H
Zilberstein S, Crites R, Watkins C, Houk J, Berthier N, Monaco J, Ginpen R, Sutton R, Koller D, Barto A, Singh S, Ravindran B
Prob. Learning
Neuroscience
Neural Network, Vision
Optimization, Character recognition
Learning theory
Robotics
Reinforcement Learning

Figure 3: Above plot shows hierarchical communities discovered by a 2-layer LGVG model. Top layer shows latent communities inferred from $\mathbf{Z}^{(2)}$) and bottom layer shows inferred communities from $\mathbf{Z}^{(1)}$. Communities from lower layer are mapped to upper layer by inspecting top-5 corresponding entries in $\mathbf{\Phi}^{(2)}$.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2008.

[2] P. Gopalan, J. M. Hofman, and D. M. Blei. Scalable recommendation with hierarchical poisson factorization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 326–335, Arlington, Virginia, United States, 2015. AUAI Press.

[3] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[4] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[5] C. Hu, P. Rai, and L. Carin. Deep generative models for relational data with side information. In *International Conference on Machine Learning*, pages 1578–1586, 2017.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] D. P. Kingma and M. Welling. Efficient gradient-based inference through transformations between bayes nets and neural nets. *arXiv preprint arXiv:1402.0480*, 2014.

[8] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[10] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, 2011.

[11] N. Mehta, L. Carin, and P. Rai. Stochastic blockmodels meet graph neural networks. *arXiv preprint arXiv:1905.05738*, 2019.

[12] K. Miller, M. Jordan, and T. Griffiths. Nonparametric latent feature models for link prediction. In *NIPS*, 2009.

[13] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *JASA*, 2001.

[14] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[15] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *NIPS*, 2016.

[16] E. W. Stacy. A generalization of the gamma distribution. *The Annals of mathematical statistics*, pages 1187–1192, 1962.

[17] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, Nov 2011.

[18] H. Zhang, B. Chen, D. Guo, and M. Zhou. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *ICLR*, 2018.

[19] M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, 2015.

A Appendix

A.1 Experiment details

Link Prediction: We evaluated our model and the baselines for link-prediction on 4 real-world benchmark graph datasets - NIPS12, Cora, Citeseer, and Pubmed . For our model, we set the gamma shape hyperparameters as 10^{-5} for the topmost layer and for subsequent layers, shape parameter is drawn as per Eq. 1. The gamma rate parameter was set as 10^{-3} for top layer, and 10^{-2} for subsequent layers (the model was mostly insensitive to the choice of the rate parameter). We used two layers in both encoder and decoder network with layers sizes (bottom and top) being 128 and 64 for Cora, Citeseer and Pubmed, and 64 and 32 for NIPS12. Model was trained on all datasets for 500 epochs. The reason of slightly different settings for evaluation of NIPS12 is extremely sparse connectivity compared to other datasets. Our evaluation scheme is similar to that of [9], and the reported scores are averaged over 10 random splits of the data. The results for link-prediction task are shown in Table 1 and Table 2, in which we have used the Average Precision (AP) and AUC-ROC scores as the metric. In addition to the models in [9], we compare our results with a fast SGVB-based implementation of EPM (which is roughly a single layered version of our model) [19].

Table 2: ROC (AUC).

Method	**NIPS12**	**Cora**	**Citeseer**	**Pubmed**
SC [17]	0.8792 ± 0.0003	0.8460 ± 0.0001	0.8050 ± 0.0001	0.8420 ± 0.0002
DW [14]	0.8058 ± 0.0000	0.8310 ± 0.0001	0.8050 ± 0.0002	0.8440 ± 0.0000
VGAE [9]	0.9029 ± 0.0031	0.9140 ± 0.0001	0.9080 ± 0.0002	0.9440 ± 0.0002
DGLFRM [11]	0.8734 ± 0.0043	0.9343 ± 0.0023	0.9379 ± 0.0032	0.9395 ± 0.0008
EPM-SGVB [19]	0.8736 ± 0.0155	0.8489 ± 0.0114	0.7714 ± 0.0181	0.8339 ± 0.0079
LGVG (this paper)	0.9100 ± 0.0084	0.9320 ± 0.0051	0.9128 ± 0.0116	**0.9601** ± 0.0017
LGVG-X (this paper)	**0.9100** ± 0.0084	**0.9524** ± 0.0049	**0.9615** ± 0.0071	0.9590 ± 0.0012

Table 3: Example of topics inferred by our model on NIPS data. For each community, authors are ranked by their strengths in respective communities. Authors belonging to multiple communities are highlighted.

Inferred topic(s)	**Authors**
Learning Th. & Optimization	Zhao J, Platt J, Bartlett P, Shawe-Taylor J, Helmke U, Hancock T, Mason L, Spence C, Campbell C, Scholkopf B
Reinforcement Learning	Singh S, Barto A, Horn D, Connolly C, Sutton R, Berthier N, Koller D, Ginpen R, Precup D, Rodriguez A
Computer Vision	Rosenfeld R, Bengio Y, **LeCun Y**, Singer Y, Isabelle J, Mato G, Turiel A, Nadal J, **Boser B**, Bengio S
Probabilistic Learning	Williams C, Jordan M, **Goldberg P**, Vivarelli F, Bishop C, Ghahramani Z, Lawrence N, Ueda N, Teh Y, Hinton G, Ng A
Neuroscience	Goldstein M, Burnod Y, Osterholtz L, Touretzky D, Burger M, de-Oliveira P, Russell S, Sumida R, Martignon L, **Goldberg P**, Principe J
Character recognition	Janow R, Lee R, Vapnik V, **LeCun Y**, Cortes C, Denker J, Sackinger E, Nohl C, Solla S, Jackel L, **Boser B**

Community discovery in NIPS12 dataset: We train both our model and VGAE with same last layer size (128). Table 3 shows communities learned by LGVG. It can be seen that some of the members appear in multiple communities (Yann LeCun, Paul Goldberg for instance appear in two different communities). In addition, our model can learn a hierarchical community structure, and we show an illustration for the same in Fig. 4. For the tree structured visualization (repeated in Appendix for ease of access), we first pick a few communities inferred from last layer embeddings ($\mathbf{Z}^{(1)}$). Then, for each community *i*, we find out the communities *j* in upper layer ($\mathbf{Z}^{(2)}$) having maximum connection weight as per $\boldsymbol{\Phi}$ ($\phi^{(2)}_{i_j}$) (top-5). We connect each community at a higher layer, to one in lower layer by edges with weights proportional to the connection strength between them. It can be seen that communities at a higher level are mixture of communities at a lower level.

A.2 Dataset details:

We consider several real world datasets, with three datasets also consisting of side information (in form of node features and node labels for a fraction of nodes), and other datasets only having node connectivity information. For the datasets with node labels, we use training label fraction as mentioned in [8]. Details of each dataset are as follows:

- **NIPS12**: The NIPS12 coauthor network [3] includes coauthorship data for all 2037 authors in NIPS vols 0-12, with 3134 edges. It has no side information.
- **Cora**: Cora network is a citation network consisting of 2708 documents. It contains sparse bag-of-words feature vectors of length 1433 for each document. These are used as node features. Cora dataset also has node labels for 140 nodes.
- **Citeseer**: Citeseer is a citation network consisting of 3312 scientific publications from six categories - agents, AI, databases, human computer interaction, machine learning and information retrieval. The side information for the dataset is the category label for each paper which is converted into a one-hot representation. This dataset also has node labels for 120 nodes. The network has total 4552 links.
- **Pubmed**: It is a citation network consisting of 19717 nodes. The dataset contains sparse bag-of-word features of length 500 for each document. Additionally, this dataset has node labels for 60 nodes. The network has total 44324 links.

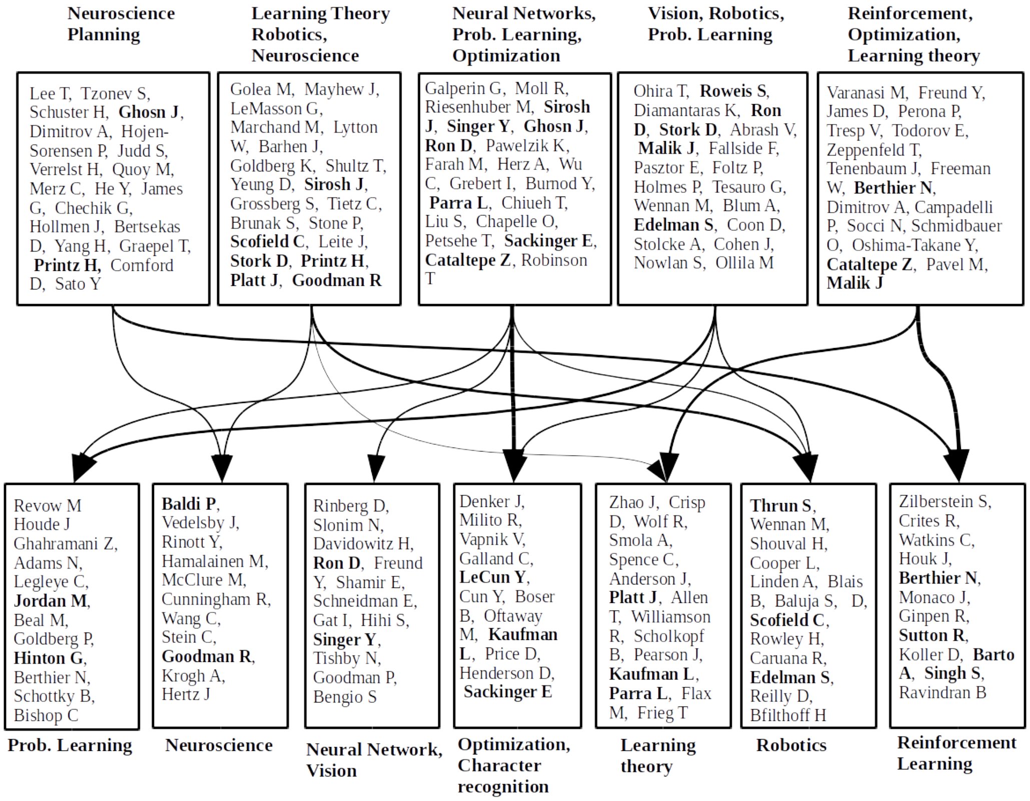

Neuroscience Planning
Learning Theory Robotics, Neuroscience
Neural Networks, Prob. Learning, Optimization
Vision, Robotics, Prob. Learning
Reinforcement, Optimization, Learning theory
Lee T, Tzonev S, Schuster H, Ghosn J, Dimitrov A, Hojen-Sorensen P, Judd S, Verrelst H, Quoy M, Merz C, He Y, James G, Chechik G, Hollmen J, Bertsekas D, Yang H, Graepel T, Printz H, Cornford D, Sato Y
Golea M, Mayhew J, LeMasson G, Marchand M, Lytton W, Barhen J, Goldberg K, Shultz T, Yeung D, Sirosh J, Grossberg S, Tietz C, Brunak S, Stone P, Scofield C, Leite J, Stork D, Printz H, Platt J, Goodman R
Galperin G, Moll R, Riesenhuber M, Sirosh J, Singer Y, Ghosn J, Ron D, Pawelzik K, Farah M, Herz A, Wu C, Grebert I, Burnod Y, Parra L, Chiueh T, Liu S, Chapelle O, Petsehe T, Sackinger E, Cataltepe Z, Robinson T
Ohira T, Roweis S, Diamantaras K, Ron D, Stork D, Abrash V, Malik J, Fallside F, Pasztor E, Foltz P, Holmes P, Tesauro G, Wennan M, Blum A, Edelman S, Coon D, Stolcke A, Cohen J, Nowlan S, Ollila M
Varanasi M, Freund Y, James D, Perona P, Tresp V, Todorov E, Zeppenfeld T, Tenenbaum J, Freeman W, Berthier N, Dimitrov A, Campadelli P, Socci N, Schmidbauer O, Oshima-Takane Y, Cataltepe Z, Pavel M, Malik J
Revow M Houde J Ghahramani Z, Adams N, Legleye C, Jordan M, Beal M, Goldberg P, Hinton G, Berthier N, Schottky B, Bishop C
Baldi P, Vedelsby J, Rinott Y, Hamalainen M, McClure M, Cunningham R, Wang C, Stein C, Goodman R, Krogh A, Hertz J
Rinberg D, Slonim N, Davidowitz H, Ron D, Freund Y, Shamir E, Schneidman E, Gat I, Hihi S, Singer Y, Tishby N, Goodman P, Bengio S
Denker J, Milito R, Vapnik V, Galland C, LeCun Y, Cun Y, Boser B, Oftaway M, Kaufman L, Price D, Henderson D, Sackinger E
Zhao J, Crisp D, Wolf R, Smola A, Spence C, Anderson J, Platt J, Allen T, Williamson R, Scholkopf B, Pearson J, Kaufman L, Parra L, Flax M, Frieg T
Thrun S, Wennan M, Shouval H, Cooper L, Linden A, Blais B, Baluja S, D, Scofield C, Rowley H, Caruana R, Edelman S, Reilly D, Bfilthoff H
Zilberstein S, Crites R, Watkins C, Houk J, Berthier N, Monaco J, Ginpen R, Sutton R, Koller D, Barto A, Singh S, Ravindran B
Prob. Learning
Neuroscience
Neural Network, Vision
Optimization, Character recognition
Learning theory
Robotics
Reinforcement Learning

Figure 4: Above plot shows a portion of hierarchical communities discovered by a 2-layer LGVG model. It can be clearly seen how lower level communities are more specific, and communities at a higher level more general (*union* of one or more lower level communities. Top layer shows latent communities inferred from $\mathbf{Z}^{(2)}$) and bottom layer shows inferred communities from $\mathbf{Z}^{(1)}$. Communities from lower layer are mapped to upper layer by inspecting top-5 corresponding entries in $\mathbf{\Phi}^{(2)}$.

[3] http://www.cs.nyu.edu/ roweis/data.html