Graph Representation Learning via Ladder Gamma Variational Autoencoders

Arindam Sarkar,[1]* **Nikhil Mehta,**[2]* **Piyush Rai**[3]
[1]Amazon India
[2]Duke University
[3]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 of its associated nodes. In addition to leveraging the representational power of multiple layers of *stochastic* variables via the ladder VAE architecture, our framework offers the following benefits: (1) Unlike existing ladder VAE architectures based on real-valued latent variables, the gamma-distributed latent variables naturally result in non-negativity and sparsity of the learned embeddings, and facilitate their direct interpretation as membership of nodes into (possibly multiple) communities/topics; (2) A novel *recognition* model for our gamma ladder VAE architecture allows fast inference of node embeddings; and (3) The framework also extends naturally to incorporate node side information (features and/or labels). Our framework is also fairly modular and can leverage a wide variety of graph neural networks as the VAE encoder. We report both quantitative and qualitative results on several benchmark datasets and compare our model with several state-of-the-art methods.

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 (Hamilton, Ying, and Leskovec 2017b). Some of the prominent prior works in this direction include stochastic blockmodels and variants (Nowicki and Snijders 2001; Miller, Jordan, and Griffiths 2009; Airoldi et al. 2008; Latouche, Birmelé, and Ambroise 2011) and, more recently, graph neural networks (Perozzi, Al-Rfou, and Skiena 2014; Kipf and Welling 2016a; Hamilton, Ying, and Leskovec 2017a). 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) (Kipf and Welling 2016a) and its variants (Hamilton, Ying, and Leskovec 2017a) 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.

However, both stochastic blockmodels as well as graph neural networks have their own shortcomings. In particular, despite providing nice interpretability for the node embeddings, the powerful variants of stochastic blockmodels such as mixed-membership blockmodels (Airoldi et al. 2008) and overlapping stochastic blockmodels (Miller, Jordan, and Griffiths 2009; Zhou 2015) 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 a 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 (Sønderby et al. 2016) typically assume dense, Gaussian distributed latent variables in each layer, the gamma-distributed embeddings in our ladder VAE architecture naturally provide sparsity and interpretability as desired for the node embeddings. These sparse, non-negative, and multilayed embeddings can be readily used for discovering overlapping community structure that allows each node to belong to potentially more

*Equal Contribution. Majority of the work was done when Arindam and Nikhil were at IIT Kanpur.

than one community (Zhou 2015). Moreover, the multilayer embeddings learned by our gamma ladder VAE model enable us to discover community structures at multiple levels of granularity (bottom layers denoting fine-grained, specialized communities, and top layers denoting coarse-grained, generic communities). Another appealing aspect is that the shrinkage property of the gamma distribution (Zhou 2015) enables learning the node embedding size (i.e., number of communities) in each layer of the hierarchy.

Besides learning interpretable node embeddings with strong predictive power for the link-prediction task, our framework allows incorporating side-information (if available) associated with each node in a principled manner. The side-information may be given in form of node features or node labels, and we model it via a nonlinear generative model that maps the node embeddings to the side information. In the case when the side-information is in form of the labels for a subset of the nodes, our framework can also be used for semi-supervised node classification (Kipf and Welling 2016a).

Finally, our framework is also accompanied by a fast inference algorithm to infer the node embeddings in each layer of the deep hierarchy. This is accomplished via a fast *recognition model* which is part of our gamma ladder VAE architecture. The recognition model is essentially based on a graph encoder such as a graph convolutional network (GCN) (Kipf and Welling 2016a) or its more recent extensions such as GraphSAGE (Hamilton, Ying, and Leskovec 2017a). In contrast, inference in other latent variable models for graphs, such as stochastic blockmodels and its extensions (Nowicki and Snijders 2001; Miller, Jordan, and Griffiths 2009; Airoldi et al. 2008; Latouche, Birmelé, and Ambroise 2011; Zhou 2015), relies on expensive MCMC or variational inference, which do not scale easily.

The key contributions of this work can be summarized as follows: (1) We proposed a gamma ladder VAE based deep generative model for graphs where each node is modeled by multiple layers of stochastic latent variables that represent a deep hierarchy of embeddings for each node; (2) The node embeddings learned by our model have direct interpretability and can be readily used for task such as inferring node memberships to (potentially multiple) communities; (3) Node side-information, if available, can be incorporated in a principled manner; (4) A novel, fast recognition model for our proposed gamma ladder VAE enables efficient inference of node embeddings.

Background

Ladder VAEs: Our model is inspired by (but has several key differences as we discuss later) the ladder variational autoencoder (ladder VAE) (Sønderby et al. 2016). The ladder VAE is a improvement over the standard VAE (Kingma and Welling 2014) to allow having multiple stochastic layers of 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 accomplishes this via an information sharing scheme between the upward deterministic pass and the downward stochastic pass during inference of latent variables. However, our proposed ladder VAE has some key differences from existing ladder VAE: (1) While existing ladder VAE are designed to model data such as images and consists of Gaussian latent variables, our ladder VAE architecture is designed to model graphs and consists of gamma latent variables, which result in non-negative, sparse node embeddings with direct interpretability; (2) As we show in Inference section, a ladder VAE with gamma latent variables is not straightforward to do inference on, since gamma random variables do not allow easy reparameterization for training (Knowles 2015); we overcome this by reparametrizing the gamma via the *generalized* gamma distribution (Stacy 1962).

Graph Encoders: Our inference network (recognition model) has a deterministic upward pass and a stochastic downward pass. For the deterministic upward pass, any nonlinear encoder for graph-structured data can be used. Graph convolutional networks (GCN) (Kipf and Welling 2016a) have recently emerged as a flexible encoder for graphs (similar in spirit to CNNs for images) which makes them an ideal choice for the upward pass in our generative framework. GCN uses a series of convolution operators to find a nonlinear deterministic embedding for each node. Our model is also fairly modular, and although we have used vanilla GCN in our architecture, any other state-of-the-art non-probabilistic graph encoders can also be used as building block for the upward pass like GraphSAGE (Hamilton, Ying, and Leskovec 2017a).

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 $\{\boldsymbol{z}_i^{(l)}\}_{\ell=1}^{L}$ and $\{\boldsymbol{z}_j^{(l)}\}_{\ell=1}^{L}$. Here, the set of latent variables $\{\boldsymbol{z}_i^{(1)}...,\boldsymbol{z}_i^{(L)}\}$ denote the latent embeddings of the node i, where $\boldsymbol{z}_i^{(l)} \in \mathbb{R}_+^{K_l}$ and K_l is the embedding size in layer ℓ. The deep hierarchy of gamma-distributed embeddings is generated as follows

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

Here, for compactness of notation, we have used a vectorized notation for the gamma distribution, so in the notation $\boldsymbol{z} \sim \text{Gamma}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{r}})$ used above, $\boldsymbol{z}, \hat{\boldsymbol{s}}, \hat{\boldsymbol{r}}$ are all vectors of same size. Note that, for $\ell = 1, 2, ..., L-1$, the shape parameter of the gamma prior depends on the embedding $\boldsymbol{z}_i^{(\ell+1)}$. Here $\boldsymbol{\Phi}^{(l)} = [\boldsymbol{\phi}_1^{(l)}, \boldsymbol{\phi}_2^{(l)}, ...\boldsymbol{\phi}_{K_l}^{(l)}]$ denotes a non-negative $K_{l-1} \times K_l$ transformation matrix, with each column $\boldsymbol{\phi}_{k_l}^{(l)} \in \mathbb{R}_+^{K_{l-1}}$ summing to one.

Interpretable node embeddings: 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* (Zhou 2015). In particular, each component of the node embedding vector $\boldsymbol{z}_i^{(l)} \in \mathbb{R}_+^{K_l}$ denotes the strength of membership of node i into one of the K_ℓ communities. Also

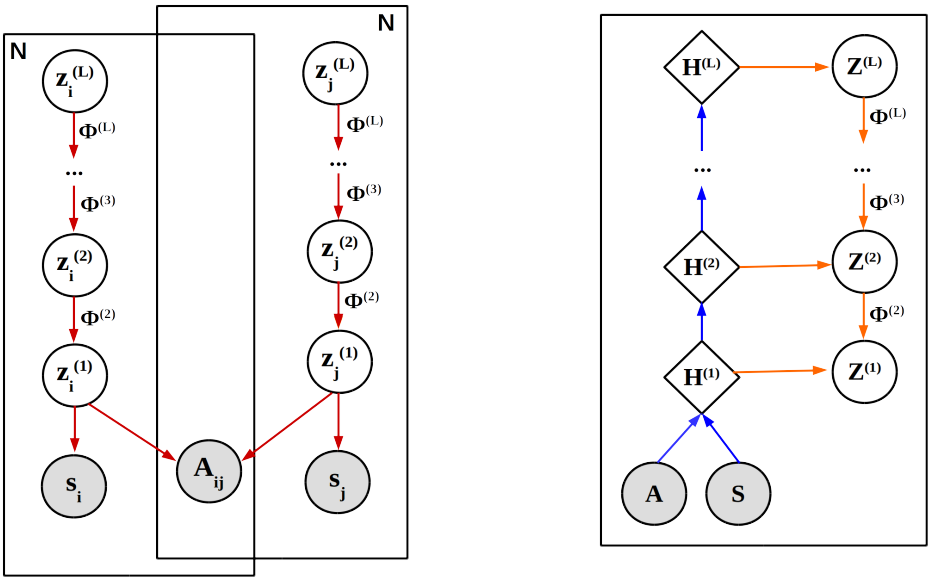

Figure 1: (Left) The decoder/generator network depicting link (A_{ij}) generation using the node embeddings. (Right) The inference/recognition network which takes the adjacency matrix $\mathbf{A}$ and the side information $\mathbf{S}$ as input. For the upward deterministic pass (blue), a graph encoder (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 $\mathbf{Z}^{(l)} = \left[\boldsymbol{z}_1^{(l)}, ..., \boldsymbol{z}_N^{(l)}\right]^T$.

note that, with such a representation, each node can potentially belong to multiple communities. Moreover, since our model learns a multilayer embedding for each node, it can infer communities at multiple layers of granularities (Hu, Rai, and Carin 2017), with bottom layers representing fine-grained communities and top layers representing coarse-grained (generic) communities.

We assume that each link A_{ij} is generated as

$$A_{ij} \sim \text{Bern}(p_{ij}); \quad \text{where} \quad p_{ij} = f(\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(1)}) \tag{2}$$

Here $f(.,.)$ can be any differentiable function which takes input two vectors to give a probability score. In our implementation, we used the cosine similarity between the community memberships of the two nodes. However, other choices are possible as well, e.g., $p_{ij} = 1 - \exp(-\boldsymbol{z}_i^{(1)\top} \mathbf{M} \boldsymbol{z}_j^{(1)})$, which can incorporate an inter-community interaction matrix $\mathbf{M} \in \mathbb{R}_+^{K_1 \times K_1}$ (Zhou 2015).

We would like to contrast our gamma ladder VAE framework with some recently proposed deep generative models for graphs, such as the variational graph autoencoder (VGAE) (Kipf and Welling 2016b), Graphite (Grover, Zweig, and Ermon 2018) and deep generative latent feature relational model (DGLFRM) (Mehta, Carin, and Rai 2019). Both VGAE and Graphite are built using a VAE customized for graph-structured data which assumes a single layer of Gaussian latent variable for each node. The DGLFRM extends the VGAE framework by assuming a stick-breaking prior for the stochastic layer to infer the latent communities of the nodes. These latent variables undergo a nonlinear transformation via a deep neural net to generate links. In contrast, our gamma ladder VAE architecture assumes multiple layers of latent variables for each node, which are directly interpretable due to the non-negativity and sparsity of gamma-distributed latent variables. Moreover, while the VGAE requires a careful tuning of the embedding size, the shrinkage property of the gamma (Zhou 2015) helps us infer the size of embedding in each layer by specifying a large-enough truncation level. In our experiments, we compare our model with VGAE and DGLFRM.

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) (Kingma and Welling 2013; 2014) to perform inference for our model. Similar to the standard VAE and ladder VAE, our inference procedure consists of a fast *recognition model* that enables inferring the latent variables of the model (in our case, these are the node embeddings in each layer of the deep hierarchy). Note that, in contrast to our deep generative model for graphs that uses a recognition model for fast inference of latent variables, other existing single-layer (Nowicki and Snijders 2001; Miller, Jordan, and Griffiths 2009; Airoldi et al. 2008; Latouche, Birmelé, and Ambroise 2011) as well as deep (Hu, Rai, and Carin 2017) stochastic block-models rely on iterative procedures, such as MCMC or variational inference. Another recently proposed latent variable model for graph, the Edge Partition Model (EPM) (Zhou 2015), assumes a single layer of gamma-distributed embed-

dings for each node and relies on MCMC inference. Fig. 1 (Right) shows the inference network used for our recognition model.

We approximate the model's true posterior $p(\mathbf{Z}|\mathbf{A},\mathbf{S})$ with a variational posterior $q(\mathbf{Z})$. We assume mean-field approximation for the variational posterior and factorize it as

$$\prod_{i=1}^{N} q(\boldsymbol{z}_i^{(L)}|\boldsymbol{h}_i^{(L)}) \prod_{\ell=1}^{L-1} q(\boldsymbol{z}_i^{(\ell)}|\boldsymbol{h}_i^{(\ell)}, \boldsymbol{z}_i^{(\ell+1)}) \tag{3}$$

The SGVB algorithm for standard VAE as well as ladder VAE leverges the reparametrization trick, which is easy for Gaussian latent variables. However, 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 (Stacy 1962), are closely related and have similar PDFs. Weibull can easily be reparameterized (Zhang et al. 2018) as follows

$$\begin{aligned} q(\boldsymbol{z}_i^{(l)}) = \text{Weibull}(\boldsymbol{\alpha}^{(l)}, \boldsymbol{\beta}^{(l)}), \quad U \sim \text{Uniform}(0,1) \\ \boldsymbol{z}_i^{(l)} = \boldsymbol{\beta}^{(l)}(-\ln(1-U))^{\frac{1}{\boldsymbol{\alpha}^{(l)}}} \end{aligned} \tag{4}$$

Our inference network consists of a nonlinear graph encoder to learn the parameters of the variational posterior as defined in Eq. 4. The encoder model is based on the recognition network used in ladder VAE (Sønderby et al. 2016) (but with Weibull approximate latent variables) that first uses a deterministic upward pass to compute the approximate likelihood contributions from the data

$$\mathbf{H}^{(1)} = \text{Graph-ENC}^{(1)}(\mathbf{A},\mathbf{S}),\ \mathbf{H}^{(l)} = \text{Graph-ENC}^{(l)}(\mathbf{H}^{(l-1)}), \tag{5}$$

$$\hat{\mathbf{S}}^{(1)} = \text{Softplus}(\mathbf{H}^{(1)}\mathbf{W}_s^{(1)}),\ ...,\ \hat{\mathbf{S}}^{(l)} = \text{Softplus}(\mathbf{H}^{(l)}\mathbf{W}_s^{(l)}), \tag{6}$$

$$\hat{\mathbf{R}}^{(1)} = \text{Softplus}(\mathbf{H}^{(1)}\mathbf{W}_r^{(1)}),\ ...,\hat{\mathbf{R}}^{(l)} = \text{Softplus}(\mathbf{H}^{(l)}\mathbf{W}_r^{(l)}), \tag{7}$$

The upward pass takes as input $\{\mathbf{A},\mathbf{S}\}$ to generate deterministic layers $\{\mathbf{H}^{(l)}\}_{l=1}^{L}$ via series of nonlinear transformations defined by Graph-ENC as shown in Eq. 7, where $\mathbf{H}^{(l)} \in \mathbb{R}^{N\times D_l}$, N is the total number of nodes and D_l is the size of l^{th} deterministic layer. We have used a graph convolution network (GCN) (Kipf and Welling 2016a) with L layers for these nonlinear transformations. The information sharing contributions $\{\hat{\mathbf{S}}^{(l)},\hat{\mathbf{R}}^{(l)}\}_{l=1}^{L}$ are computed using Eq. 7, where $\mathbf{W}_s^{(l)} \in R^{D_l\times K_l}$, $\mathbf{W}_r^{(l)} \in R^{D_l\times K_l}$, $\hat{\mathbf{S}}^{(l)} \in \mathbb{R}_+^{N\times K_l}$ and $\hat{\mathbf{R}}^{(l)} \in \mathbb{R}_+^{N\times K_l}$ with K_l being the size of the l^{th} stochastic layer. The upward pass is followed by a stochastic downward pass to compute the generative distributions and their variational parameters. The variational distributional is defined as $\prod_{i=1}^{N} q(\boldsymbol{z}_i^{(L)}|\boldsymbol{h}_i^{(L)}) \prod_{l=1}^{L-1} q(\boldsymbol{z}_i^{(l)}|\boldsymbol{h}_i^{(l)}, \boldsymbol{z}_i^{(l+1)})$.

The downward pass can be performed recursively as

$$\begin{aligned} q(\boldsymbol{z}_i^{(L)}|\boldsymbol{h}_i^{(L)}) &= \text{Weibull}(\boldsymbol{\alpha}_i^{L}, \boldsymbol{\beta}_i^{L}), \\ q(\boldsymbol{z}_i^{(\ell)}|\boldsymbol{h}_i^{(l)}, \boldsymbol{z}_i^{(l+1)}) &= \text{Weibull}(\boldsymbol{\alpha}_i^{l}, \boldsymbol{\beta}_i^{\ell}), \\ q(\boldsymbol{z}_i^{(1)}|\boldsymbol{h}_i^{(1)}, \boldsymbol{z}_i^{(2)}) &= \text{Weibull}(\boldsymbol{\alpha}_i^{1}, \boldsymbol{\beta}_i^{1}) \end{aligned} \tag{8}$$

where $\boldsymbol{\alpha}_i^l = \hat{\boldsymbol{s}}_i^{(L)}$, $\boldsymbol{\beta}_i^l = \hat{\boldsymbol{r}}_i^{(L)}$ if $l = L$, otherwise $\boldsymbol{\alpha}_i^l = \hat{\boldsymbol{s}}_i^{(l)} + \boldsymbol{\Phi}^{l+1}\boldsymbol{z}_i^{l+1}$, $\boldsymbol{\beta}_i^l = \hat{\boldsymbol{r}}_i^{(l)}$. Following the SGVB training scheme (Kingma and Welling 2013), 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 $\boldsymbol{z}_n^{(\ell+1)}$.

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{E}\big[\ln p(A_{ij}|\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(1)})\big] - \\ \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\Big[\ln q(\boldsymbol{z}_n^{(l)}) - \ln p(\boldsymbol{z}_n^{(l)})\Big] \end{aligned} \tag{9}$$

where all expectations are taken w.r.t the variational distribution q. The second term in the ELBO is the KL-Divergence between the posterior Weibull distribution and the prior Gamma distribution which has an analytic form (Bauckhage 2014) given by:

$$\begin{aligned} \text{KL}(\text{Weibull}(\alpha,\beta)||\text{Gamma}(\hat{s},\hat{r})) = \frac{\gamma\hat{r}}{k} + \ln\alpha + \\ \hat{r}\beta\Gamma(1+\frac{1}{\alpha}) + \ln\Gamma(\hat{s}) - \hat{s}\ln\beta - \beta - 1 - \hat{s}\ln\hat{r} \end{aligned} \tag{10}$$

Related Work

A prominent class of methods for modeling graphs is based on associating a latent variable with each node of the graph. Stochastic blockmodels (Nowicki and Snijders 2001; Kemp et al. 2006) and their variants such as mixed-membership stochastic blockmodels (Airoldi et al. 2008) and overlapping stochastic blockmodels (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011) belong to this class of models. These models have proven to be highly effective in discovering the latent structure in the graph, e.g., communities, and also yield strong link prediction performance. However, these models only assume a single layer latent representation for each node and simple link generation models (e.g., a bilinear function of the embeddings of the associated nodes). Moreover, inference in these models is based on iterative methods such as MCMC and variational inference, and therefore tends to be slow. Another recent work in this direction is the Edge Partition Model (EPM) (Zhou 2015) which is another overlapping stochastic blockmodel with gamma-distributed embeddings for each node. The use of gamma-distributed embeddings in EPM is similar in spirit to our approach. However, EPM is a single layer model and relies on expensive MCMC procedure to infer the node embeddings. In contrast, our model is a deep generative model with gamma-distributed sembeddings in each layer, and has an efficient recognition model to infer the node embeddings.

Deep generative models for graphs using graph neural networks (Scarselli et al. 2009) as their building blocks have recently been proposed. However, the existing models, such as graph variational autoencoder (Kipf and Welling 2016b) and Graphite (Grover, Zweig, and Ermon 2018) are based

on vanilla VAE architectures, consisting of a single layer of stochastic latent variables which undergo multiple layers of deterministic transforms (e.g., via GCN). Moreover, these deep generative models assume Gaussian embeddings, that do not have the nice interpretability as offered by our gamma ladder VAE architecture. Another recent development is the deep generative model for graphs (Hu, Rai, and Carin 2017), with multiple layers of binary-valued stochastic variables based on a sigmoid belief network. However, inference in this model is expensive and relies on an MCMC procedure.

Other recently proposed prominent methods for learning representations (embeddings) for the nodes in the graph include methods that are based on the idea of random walks and neighborhood preservation, e.g., DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and node2vec (Grover and Leskovec 2016). However, these methods do not learn a deep hierarchical representations unlike our model and the learned embedding are not interpretable. Some other related methods are FastGCN (Chen, Ma, and Xiao 2018) which speeds up GCN by proposing a batched training scheme, and Graph Attention Networks (Veličković et al. 2017) which leverages an attention based architecture for node classification on graph-structured data.

Our work is inspired by the success of deep generative models, in particular models like ladder VAE (Sønderby et al. 2016) that effectively allow for multiple layers of *stochastic* variables (unlike vanilla VAEs which only assume a single layer of latent variables) to leverage the power of hierarchical latent representation. Although ladder VAEs have been successful for modeling data such as images (Sønderby et al. 2016) and text (Zhang et al. 2018), our work provides the first such construction of such models for graph-structured data. As we will show, Gaussian VAEs/Ladder VAEs are not ideal for learning interpretable graph embeddings, which is a major motivation for our work.

Experiments

We evaluate our model on several synthetic and real datasets and compare with various state-of-the-art baselines. 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**. We report both qualitative results (e.g., inferred node embeddings/communities) and quantitative results on link-prediction. Our baselines consist of Spectral Clustering (SC) (Tang and Liu 2011), DeepWalk (DW) (Perozzi, Al-Rfou, and Skiena 2014), the Variational Graph Autoencoder (VGAE) (Kipf and Welling 2016b), the Deep Generative Latent Feature Relational model (DGLFRM) (Mehta, Carin, and Rai 2019), and single-layer/deep probabilistic models, namely the Edge Partition Model (EPM) (Zhou 2015). Unless explicitly stated, for link prediction and community discovery tasks, we use 85% adjacency matrix used for training, 5% for validation and 10% for testing. Additional results are included in a longer version of the paper available in arXiv [1]. For the node-community visualizations, we follow node-community reordering scheme used in (Zhou 2015).

[1] https://arxiv.org/

Qualitative Results on Synthetic Datasets

To assess our model's capability for inferring the community structure within a graph and the nature of node embeddings learned, we apply our model on two synthetic datasets, each of which consists of 150 nodes. The first dataset consists of 10 non-overlapping communities and the second dataset consists of 10 overlapping communities. The results are shown in Fig. 2. As these plots show, our model learns node embeddings that are readily interpretable whereas VGAE learns dense embeddings that do not have an easy interpretability. Moreover, our model is also able to infer the number of communities (we specified a large truncation level of 16 and the model correctly inferred around 10 active communities), whereas in case of VGAE with dense embeddings, this parameter needs to be tuned.

Quantitative Results on Real Datasets

Link-Prediction: We next evaluate our model and the baselines for link-prediction on 4 real-world benchmark graph datasets - NIPS12 (2037 nodes) [2], Cora (2361 nodes), Citeseer (3312 nodes), and Pubmed (19717 nodes) (Kipf and Welling 2016a). For our model, we set the gamma shape hyperparameter of top-layer as 10^{-5} and for subsequent layers, shape parameter is drawn as per 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. All datasets were trained 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 (Kipf and Welling 2016b), and the reported scores are averaged over 10 random splits of the data. The results for link-prediction task are shown in Table 1 in which we have used the Average Precision (AP). In addition to the models in (Kipf and Welling 2016b), we compare our results with a fast SGVB-based implementation of Edge Partition Model (which is roughly a single layered version of our model) (Zhou 2015).

Benefit of Deep Hierarchy of Latent Variables: Our gamma ladder VAE based model uses multiple layers of stochastic latent variables, in addition to multiple graph-convolution units from GCN-architecture. While multi-layered graph convolutions aid in capturing *n*-hop neighborhoods, multiple stochastic layers allows the model to capture hierarchical structure present in data. Moreover, we observed that the performance of our model improved steadily as we increased the number of latent variables in the model. We plot the ROC and AP scores with different number of layers for the NIPS12 dataset in Fig.3(b). In this experiment, the number of hidden layers ranged from 1-5 and the number of nodes in each layer ranged from 4-64.

Performance with Varying Fraction of Training Data: We also compare our model with VGAE in a setting where we vary the percentage of missing data in the graph. As

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

Table 1: Average Precision (AP).

Method	NIPS12	Cora	Citeseer	Pubmed
SC (Tang and Liu 2011)	0.9022 ± 0.0003	0.8850 ± 0.0000	0.8500 ± 0.0001	0.8780 ± 0.0001
DW (Perozzi, Al-Rfou, and Skiena 2014)	0.8634 ± 0.0000	0.8500 ± 0.0000	0.8360 ± 0.0001	0.8410 ± 0.0000
VGAE (Kipf and Welling 2016b)	0.9111 ± 0.0025	0.9260 ± 0.0001	0.9200 ± 0.0002	0.9470 ± 0.0002
DGLFRM (Mehta, Carin, and Rai 2019)	0.9005 ± 0.0027	0.9376 ± 0.0022	0.9438 ± 0.0073	0.9497 ± 0.0035
EPM-SGVB (Zhou 2015)	0.9086 ± 0.0129	0.8666 ± 0.0109	0.8259 ± 0.0172	0.8600 ± 0.0047
LGVG (Ours)	0.9260 ± 0.0068	0.9254 ± 0.0068	0.9130 ± 0.0112	0.9545 ± 0.0024
LGVG-X (Ours)	**0.9260** ± 0.0068	**0.9502** ± 0.0061	**0.9624** ± 0.0067	**0.9559** ± 0.0017

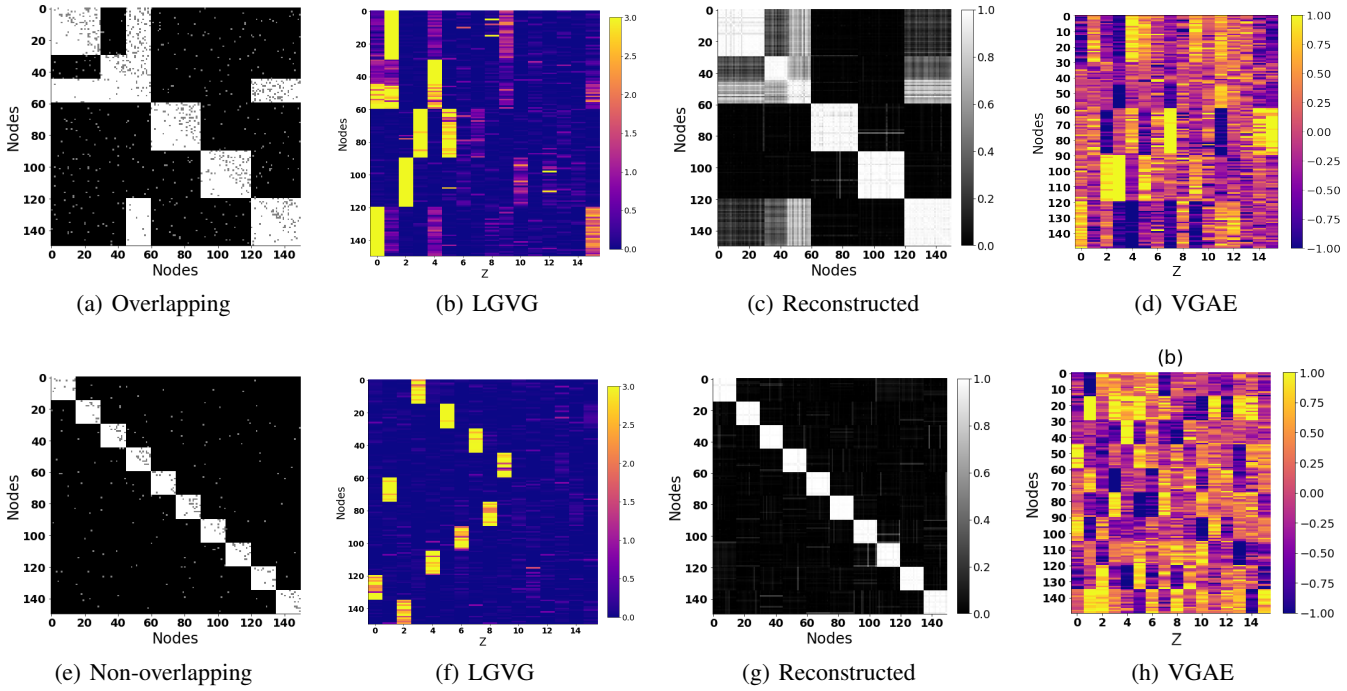

Figure 2: (a-d) Experiment on the overlapping synthetic dataset. (a) The training adjacency matrix (Black, white and Gray denote no-link, link and the 15% masked respectively). (b) The node embeddings learned at the last layer by the LGVG. (c) The reconstructed graph using the LGVG model indicating probability of link between nodes (white (black) suggest high probability of link (no-link)). (d) The node embeddings learned by the VGAE model. (e-h) A similar experiment on the non-overlapping synthetic dataset. (e) The training adjacency matrix. (f) The node embeddings learned by the LGVG. (g) The graph reconstruction using LGVG model. (h) The node embedding learned by the VGAE model.

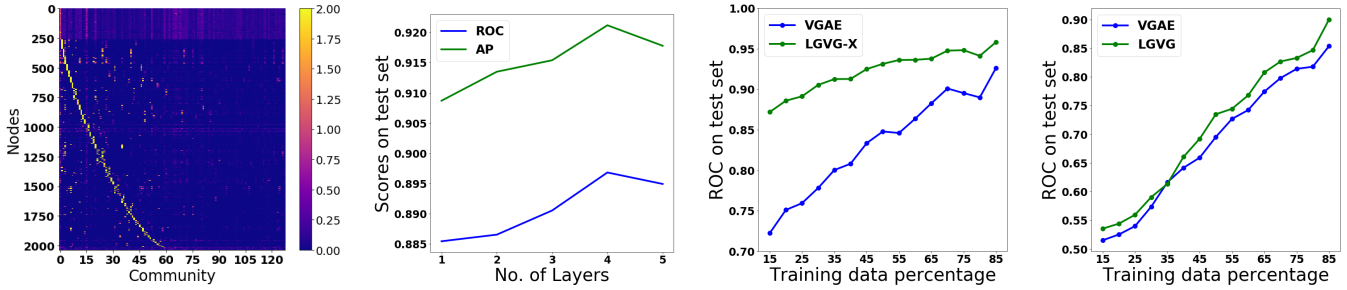

Figure 3: From left. (a) Plot of $\mathbf{Z}^{(1)}$ showing the inferred communities for NIPS12 data. (b) Plot shows test ROC and AP scores of our model on NIPS12 data with varying number of stochastic layers. (c-d) Comparison of our model with VGAE on Cora dataset, with varying percentage of adjacency matrix available at training. (c) With node side-information. (d) Without node side-information.

Table 2: 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**

shown in Fig. 3 (c-d), our model performs significantly better than VGAE when the percentage of missing data is high. The better performance of our model is attributed to (1) The gamma ladder VAE construction which can learn more powerful representations with higher predictive performance, and (2) The model's ability to incorporate the node side-information more effectively.

Model complexity: In this work, we use a batch implementation of Encoder and Decoder, which scales quadratically in the number of nodes, however, it can be extended to use mini-batch optimization in the SGVB algorithm, as well as in the encoder (e.g., by replacing GCN by GraphSAGE).

Qualitative Results on NIPS12 Data

We also conduct a qualitative analysis experiment on NIPS12 co-authorship data to examine the community structure discovered by our model. We train both our model and VGAE with same last layer size (128). The inferred last layer embeddings are shown in Fig 3. It can be seen that the model learns sparse embeddings for each authors (reflecting each author's memberships in a small number of communities). Note that VGAE learns dense embeddings that are not directly interpretable and require an additional clustering step.

Table 2 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, 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 $\mathbf{\Phi}$ ($\phi_{ij}^{(2)}$) (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.

Conclusion

We have presented a novel, gamma ladder VAE model for graph-structured data, that enables us to infer multilayered

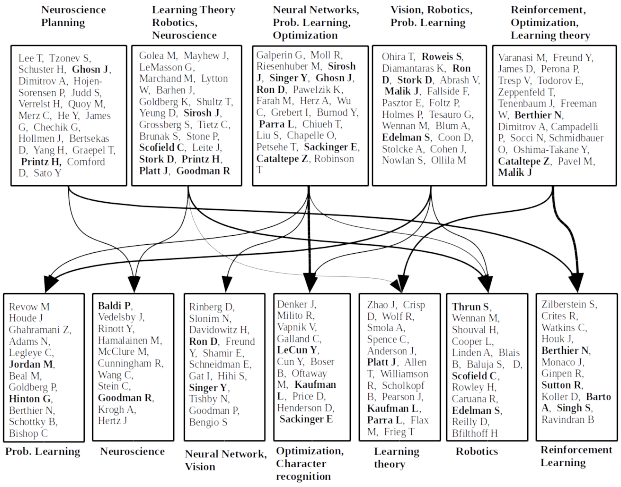

Figure 4: Above plot shows hierarchical communities discovered by a 2-layer LGVG model. Top layer shows latent communities inferred from $\mathbf{Z}^{(2)}$) and bottom layer shows inferred communities from $\mathbf{Z}^{(1)}$. Communities from lower layer are mapped to upper layer by inspecting top-5 corresponding entries in $\mathbf{\Phi}^{(2)}$.

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 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.

Acknowledgements

Piyush Rai acknowledges support from Google AI/ML Faculty Award and Visvesvaraya Young Faculty Fellowship by MeiTY, India.

References

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

Bauckhage, C. 2014. Computing the kullback-leibler divergence between two generalized gamma distributions. *CoRR* abs/1401.6853.

Chen, J.; Ma, T.; and Xiao, C. 2018. Fastgcn: Fast learning with graph convolutional networks via importance sampling. *CoRR* abs/1801.10247.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Grover, A.; Zweig, A.; and Ermon, S. 2018. Graphite: Iterative generative modeling of graphs. *arXiv preprint arXiv:1803.10459*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017a. Inductive representation learning on large graphs. In *NIPS*.

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

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

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the national conference on artificial intelligence*.

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

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

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

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

Knowles, D. A. 2015. Stochastic gradient variational Bayes for gamma approximating distributions. *ArXiv e-prints*.

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

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

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

Miller, K.; Griffiths, T.; and Jordan, M. 2009. Nonparametric latent feature models for link prediction. *NIPS*.

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

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

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

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* 20(1):61–80.

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

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

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

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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

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

Supplementary material: Graph Representation Learning via Ladder Gamma Variational Autoencoders

Anonymous Author(s)

Quantitative experiment: AUC-ROC and AP

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. All datasets were trained 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 (Kipf and Welling 2016b), and the reported scores are averaged over 10 random splits of the data. The results for link-prediction task are shown in Table 2 and Table 3, in which we have used the Average Precision (AP) and AUC-ROC scores as the metric. In addition to the models in (Kipf and Welling 2016b), we compare our results with a fast SGVB-based implementation of EPM (which is roughly a single layered version of our model). (Zhou 2015).

Qualitative: Visualizations on NIPS12

Fig 1 shows final layer embeddings from LGVG (left) and VGAE (right). Here we have followed node-community reordering scheme used in (Zhou 2015).

Semi-supervised Node classification

We also experiment with end-to-end semisupervised node classification, with same train, validation and test splits as GCN (Kipf and Welling 2016a). Note that even though we focus on link prediction and community discovery, we perform favorably against GCN on the task of semi-supervised node-classification. We have used regularization on linear layers used for information sharing between recognition and generative network. Weight decay parameter has been set to to 10^{-4} for Citeseer and 10^{-5} for Cora and Pubmed.

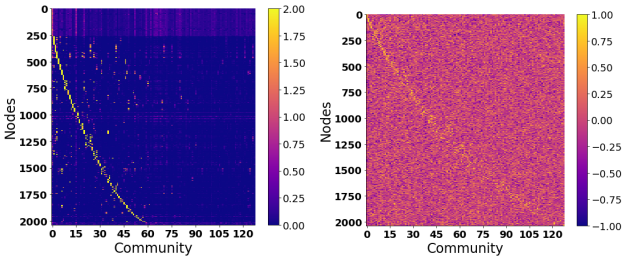

Figure 1: (Left) shows inferred communities by our model. (Right) shows last layer embeddings of VGAE on NIPS12 dataset.

Table 1: Summary of semi-supervised node classification accuracy (in percent). Baseline numbers are from (Kipf and Welling 2016a)

Method	Cora	Citeseer	Pubmed
ManiReg	59.5	60.1	70.7
SemiEmb	59.0	59.6	71.1
LP	68.0	45.3	63.0
DeepWalk	67.2	43.2	65.3
ICA	75.1	69.1	73.9
Planetoid	75.7	64.7	77.2
GCN	**81.5**	**70.3**	79.0
LGVG (this paper)	80.5	**70.3**	**79.9**

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 (Kipf and Welling 2016a). Details of each dataset are as follows:

- **NIPS12**: The NIPS12 coauthor network [1] 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

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

Table 2: Average Precision (AP).

Method	NIPS12	Cora	Citeseer	Pubmed
SC (Tang and Liu 2011)	0.9022 ± 0.0003	0.8850 ± 0.0000	0.8500 ± 0.0001	0.8780 ± 0.0001
DW (Perozzi, Al-Rfou, and Skiena 2014)	0.8634 ± 0.0000	0.8500 ± 0.0000	0.8360 ± 0.0001	0.8410 ± 0.0000
VGAE (Kipf and Welling 2016b)	0.9111 ± 0.0025	0.9260 ± 0.0001	0.9200 ± 0.0002	0.9470 ± 0.0002
DGLFRM (Mehta, Carin, and Rai 2019)	0.9005 ± 0.0027	0.9376 ± 0.0022	0.9438 ± 0.0073	0.9497 ± 0.0035
EPM-SGVB (Zhou 2015)	0.9086 ± 0.0129	0.8666 ± 0.0109	0.8259 ± 0.0172	0.8600 ± 0.0047
LGVG (this paper)	0.9260 ± 0.0068	0.9254 ± 0.0068	0.9130 ± 0.0112	0.9545 ± 0.0024
LGVG-X (this paper)	**0.9260** ± 0.0068	**0.9502** ± 0.0061	**0.9624** ± 0.0067	**0.9559** ± 0.0017

Table 3: ROC (AUC).

Method	NIPS12	Cora	Citeseer	Pubmed
SC (Tang and Liu 2011)	0.8792 ± 0.0003	0.8460 ± 0.0001	0.8050 ± 0.0001	0.8420 ± 0.0002
DW (Perozzi, Al-Rfou, and Skiena 2014)	0.8058 ± 0.0000	0.8310 ± 0.0001	0.8050 ± 0.0002	0.8440 ± 0.0000
VGAE (Kipf and Welling 2016b)	0.9029 ± 0.0031	0.9140 ± 0.0001	0.9080 ± 0.0002	0.9440 ± 0.0002
DGLFRM (Mehta, Carin, and Rai 2019)	0.8734 ± 0.0043	0.9343 ± 0.0023	0.9379 ± 0.0032	0.9395 ± 0.0008
EPM-SGVB (Zhou 2015)	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

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.

Community discovery with VGAE

Our model being able to produce sparse, positive representations, does implicit community discovery as part of learning the representation for a given task like link prediction.

We perform the following experiment in an attempt to demonstrate the effectiveness of our model vs. a model like VGAE which uses real embeddings.

We use k-means to find clusters for NIPS12 (3134 authors) co-authorship data on the node embeddings learned using VGAE. We show results by using two different community sizes (K). We try to interpret the k-means results using inferred communities by our model (which we also show for quick reference as table 5). Table 4 shows the position of selected the members of communities inferred by our model in the k-means clusters. While k-means manages to group some relevant authors together, it breaks some groups which were coherent in the communities inferred by our model. Note that k-means has no strength indicator for community membership, and also can't have overlapping communities unlike our model. Only meaningful clusters from k-means are shown, further filtered to have same authors as in communities inferred by our model for comparison.

References

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

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

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

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

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

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

Table 4: 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	Burger M, Burnod Y, Martignon L, Osterholtz L, Precup D, Zhao J
Cluster 2	Bengio Y, Boser B, Cortes C, Denker J, Jackel L, Janow R, LeCun Y, Lee R, Nohl C, Sackinger E, Scholkopf B, Shawe-Taylor J, Solla S, Vapnik V
Cluster 3	Principe J
Cluster 4	Barto S, Bengio S, Berthier N, Conolly C, Ginpen R, Isabella J, Mato G, Nadal J, Singer Y, Sumida R, Sutton R, Turiel A
Cluster 5	Barlett P, Campbell C, Helmke U, Hancock T, Horn D, Mason L, Rosenfeld R, Spence C, Touretzky D, de-Oliveira P
Cluster 6	Bishop C, Ghahramani Z, Goldberg P, Goldstein M, Koller D, Ng A, Rodriguez A, Platt J, Russel S, Teh Y, Vivarelli F, Williams C, Ueda N
Cluster (K=24)	Authors
Cluster 1	Campbell C, Helmke U, Scholkopf B, Platt J, Shawe-Taylor J, Spence C, Zhao J
Cluster 2	Barto A, Berthier N, Burnod Y, Connolly C, Ginpen R, Martignon L, Sutton R, Singh S
Cluster 3	Boser B, Cortes C, Denker J, Jackel L, Janow R, LeCun Y, Lee R, Nohl C, Sackinger E, Solla S, Vapnik V
Cluster 4	Bengio S, Bengio Y, Isabelle J, Koller D, Ng A, Rodriguez A, Russell S, Singer Y
Cluster 5	Ghahramani Z, Hinton G, Jordan M, Lawrence N, Teh Y, Ueda N,
Cluster 7	Bishop C, Goldberg P, Williams C, Vivarelli F

Table 5: Example of topic(s) inferred by our model on NIPS data. For each community, authors are ranked by their strengths in respective communities. Authors belong 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**