CONDITIONAL LABELED GRAPH GENERATION WITH GANS

Shuangfei Fan
Virginia Tech
sophia23@vt.edu

Bert Huang
Virginia Tech
bhuang@vt.edu

ABSTRACT

As a new way to train generative models, *generative adversarial networks* (GANs) have achieved considerable success in image generation, and this framework has also recently been applied to data with graph structures. We identify the drawbacks of existing deep frameworks for generating graphs, and we propose labeled-graph generative adversarial networks (LGGAN) to train deep generative models for graph-structured data with node labels. We test our model on various types of graph datasets, such as collections of citation networks and protein graphs. Experiment results show that our model can generate diverse labeled graphs that match the structural characteristics of the training data and outperforms all baselines in terms of quality, generality, and scalability.

1 INTRODUCTION

Graphs are powerful complex data structures that can describe collections of related objects. Such collections could be atoms forming molecular graphs, users connecting on online social networks, and papers connected by citations. The connected objects, or nodes, may be of different types or classes. Methods that reason about this flexible and rich representation can empower analyses of important, complex real-world phenomena. It is important to be able to learn generative models for the distributions of graphs, which can encode knowledge about the nature of the objects represented by graphs. In this paper, we introduce a method that learns generative models for labeled graphs in which the nodes and the graphs may have categorical labels.

Recently Goodfellow et al. (2014) described generative adversarial networks (GANs), which have been widely explored in computer vision and natural language processing (Zhang et al., 2017; Yu et al., 2017) for generating realistic images and text, as well as performing tasks such as style transfer. GANs are composed of two neural networks. The first is a generator network that learns to map from a latent space to the distribution of the target data, and the second is a discriminator network that tries to distinguish real data from candidates synthesized by the generator. Those two networks compete with each other during training and each improve based feedback from the other. The success of this general GAN framework has proven it to be a powerful tool for learning the distributions of complex data. Motivated by the power of GANs, researchers have used it for generating graphs too. Bojchevski et al. (2018) proposed NetGAN, which uses the GAN framework to generate random walks on graphs. De Cao & Kipf (2018) proposed MolGAN, which generate molecular graphs using the combination of a GAN framework and a reinforcement learning objective. However, there are many limitations of existing methods, such as the generality to graphs with different structures and scalability to different sized graphs. Furthermore, they are unable to generate graphs with node labels, which is a critical feature of some graph-structured data.

Building on these advances, we propose *labeled graph generative adversarial network* (LGGAN), a deep generative model trained using the AC-GAN framework (Odena et al., 2017) to generate graph-structured data with node labels. LGGAN can be used to generate various kinds of graph-structured data, such as citation graphs, knowledge graphs, and protein graphs. Specifically, the generator in an LGGAN generates an adjacency matrix as well as labels for the nodes, and its discriminator uses a jumping knowledge network (Xu et al., 2018) to identify real graphs using adaptive, structure-aware higher-level graph features and also predict the graph labels. Our approach is the first deep generative method that addresses the generation of labeled graph-structured data. In experiments,

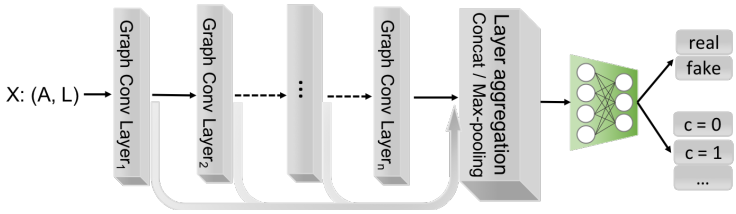

Figure 1: The structure of the discriminative model of LGGAN.

we demonstrate that our model can generate realistic graphs that preserve important properties from the training graphs. We evaluate our model on various datasets with different graph types—such as ego networks and proteins—and with different sizes. Our experiments demonstrate that LGGAN effectively learns distributions of different graph structures and that it can scale up to generate large graphs without losing much quality.

2 MODEL

In this section, we introduce LGGAN and how it trains deep generative models for graph-structured data with node labels. For the GAN framework, we adopted the AC-GAN (Odena et al., 2017) framework since it incorporates the class information that allows it to learn a better embedding and to propagate that information to the generator.

2.1 ARCHITECTURE

LGGAN consists of two main components: a generator G and a discriminator D. The generator G takes a sample from a prior distribution and generates a labeled graph g represented by an adjacency matrix A and a node label matrix L. The discriminator D then trains to distinguish samples from the dataset and generator.

Generator LGGAN's generative model uses a multi-layer perceptron (MLP) to produce the graph. The generator G takes a random vector z sampled from a standard normal distribution and outputs two matrices: (1) $L \in R^{N \times C}$, which is a one-hot vector that defines the node labels; and (2) the adjacency matrix $A \in R^{N \times N}$, which defines the connections among nodes in graphs. The architecture uses a fixed *maximum* number of nodes N, but it is capable of generating structures of fewer nodes.

Discriminator The discriminator D takes a graph sample as input (represented by an adjacency matrix A and a node label matrix L) and outputs a scalar value and also a one-hot vector for graph label. For the discriminator, we uses Jumping Knowledge Networks (Xu et al., 2018) (JK-Net). The JK-Net is comprised by a series of graph convolution layers and a layer aggregation operator to integrate useful information from each layer for learning more powerful graph representations. JK-Net with layer aggregation have been shown to extract more adaptive and structure-aware representations compared to other representation learning methods such as graph convolutional networks (GCN) (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017) and graph attention networks (GAT) (Veličković et al., 2018). For our discriminator, we use GCN as the base model.

After n layers of propagation via graph convolutions, we aggregate the outputs $H^{(l)}$ from each layer with an aggregation function agg, such as concatenation and max-pooling. We then concatenate the aggregated matrix with the node label matrix L and outputs Z_g as the final representation we learned for the graph g.

$$Z_g = f(A, L) = \left[\text{agg}\left(H^{(1)}, \ldots, H^{(n)}\right); L\right] \tag{1}$$

The representation Z_g of the graph will further be processed by a linear layer to produce the outputs of the discriminator: the graph-level scalar probability of the input being real data. Also there will be a classifier to predict the category that the graph belongs to with a one-hot vector c. The structure of this discriminative model is showing in Figure 1.

2.2 TRAINING

GANs (Goodfellow et al., 2014) train via a min-max game with two players competing to improve themselves. In theory, the method converges when it reaches a Nash equilibrium, where the samples produced by the generator matches the data distribution. However, this process is highly unstable and often results in problems such as mode collapse (Goodfellow, 2016). To deal with the most common problems in training GAN, such as mode collapse and unstable training, we use the Wasserstein GAN (Salimans et al., 2016) with a gradient penalty. We also adopt several techniques such as feature matching and minibatch discrimination that were shown to encourage convergence and help avoid mode collapse.

3 EXPERIMENTS

In this section, compare LGGAN with other graph generation methods to demonstrate its ability to generate high-quality labeled graphs in diverse settings.

3.1 BASELINES

We compare our model against various traditional generative models for graphs such as the Erdös-Rényi model (E-R) (Erdös & Rényi, 1959), the Barabási-Albert (B-A) model (Albert & Barabási, 2002), and mixed-membership stochastic block models (MMSB) (Airoldi et al., 2008). Then we also compare with some recently proposed deep graph generative models such as the DeepGMG (Li et al., 2018) and GraphRNN (You et al., 2018). Few current approaches are designed to generate labeled graphs. One exception is MolGAN (De Cao & Kipf, 2018), which is designed to generate molecular graphs and needs specialized evaluation methods specific to that task, so we do not compare against it. We also do not compare with NetGAN (Bojchevski et al., 2018) since its framework is constrained to learn to generate a single new graph from a single training graph.

3.2 DATASETS

We perform experiments on different kinds of datasets with varying sizes and characteristics.

Citation graphs We first used the Cora and Citeseer datasets (Sen et al., 2008) to test on scientific citation networks . To test the scalability of LGGAN, we extracted different subsets with different graph sizes by constraining the number of nodes in graph $|V|$. For small datasets (denoted cora_small and citeseer_small), we extract two-hop and three-hop ego networks with $30 \leq |V| \leq 50$. For the large datasets (denoted cora and citeseer), we extract three-hop ego networks with $150 \leq |V| \leq 200$. For the graph label of the citation ego networks, we set it to be the node label of the center node.

Protein graphs We also test on multiple collections of protein molecular graphs. The ENZYMES dataset consists of 600 enzymes (Schomburg et al., 2004). Each enzyme in the dataset is labeled with one of the six enzyme commission (EC) code top-level classes. The protein dataset includes proteins from the dataset of enzymes and non-enzymes created by Dobson & Doig (2003). There are two graph labels: enzymes and non-enzymes.

3.3 EVALUATION

To evaluate the quality of the generated graphs, we follow the approach used by You et al. (2018): we compare a distribution of generated graphs with that of real ones by measuring the *maximum mean discrepancy* (MMD) (Gretton et al., 2012) of graph statistics, capturing how close their distributions are. We use four graph statistics to evaluate the generated graphs: degree distribution, clustering coefficient distribution, node-label distribution, and average orbit count statistics.

3.4 COMPARING WITH OTHER METHODS

We compare LGGAN to other methods for generating graphs—both traditional generative models such as E-R, B-A, and MMSB as well as deep generative models that were proposed recently, such as GraphRNN and DeepGMG. DeepGMG cannot be used to generate large graphs due to its high computational complexity, so the results of DeepGMG on large graph datasets are not available.

Table 1: Comparison of LGGAN and other generative models on different graph structured data using MMD evaluation metrics.

	Cora_small				Citeseer_small				Cora			
	Degree	Clustering	Orbit	Label	Degree	Clustering	Orbit	Label	Degree	Clustering	Orbit	Label
E-R	0.68	0.94	0.48	N/A	0.63	0.86	0.12	N/A	0.88	1.45	0.27	N/A
B-A	0.31	0.53	0.11	N/A	0.37	0.18	0.11	N/A	0.54	1.06	0.16	N/A
MMSB	0.21	0.68	0.07	0.48	**0.17**	0.50	0.11	0.32	**0.12**	0.68	0.09	0.49
DeepGMG	0.34	0.44	0.27	N/A	0.27	0.36	0.20	N/A	-	-	-	-
GraphRNN	0.26	0.38	0.39	N/A	0.19	0.20	0.39	N/A	0.20	0.46	0.11	N/A
LGGAN	**0.13**	**0.08**	**0.03**	**0.11**	**0.17**	**0.13**	**0.04**	**0.09**	0.15	**0.21**	**0.06**	**0.009**
	Protein				ENZYMES				Citeseer			
	Degree	Clustering	Orbit	Label	Degree	Clustering	Orbit	Label	Degree	Clustering	Orbit	Label
E-R	0.31	1.06	0.28	N/A	0.38	1.26	0.08	N/A	0.82	1.57	**0.06**	N/A
B-A	0.93	0.88	0.05	N/A	1.17	1.08	0.51	N/A	0.32	1.04	0.08	N/A
MMSB	0.46	1.05	0.21	0.01	0.55	1.08	0.05	0.92	0.08	0.50	0.11	0.32
DeepGMG	0.96	0.63	0.16	N/A	0.43	0.38	0.08	N/A	-	-	-	-
GraphRNN	**0.04**	0.18	0.06	N/A	**0.06**	0.20	0.07	N/A	**0.20**	1.15	0.14	N/A
LG-GAN	0.18	**0.15**	**0.02**	**0.005**	0.09	**0.17**	**0.03**	**0.01**	0.25	**0.12**	**0.06**	**0.15**

For each method, we measure three aspects. The first is the quality of the generated graphs, which should be able to mimic typical topology of the training graphs. The second is the generality, where a good generative model should be able to generalize to different and complex graph-structured data. Then the last aspect is the scalability, where we want the model to be able to scale up to generate large networks instead of being restricted to relatively small graphs, such as molecules.

Table 1 lists results from our comparison. LGGAN achieves the best performance on all datasets, with 90% decrease of MMD on average compared with traditional baselines, and 30% decrease of MMD compared with the state-of-the-art deep learning baseline GraphRNN. Although GraphRNN performs well on the two smaller protein-related datasets, ENZYMES and protein, it does not maintain the same performance on large datasets, such as cora and citeseer. Also in order to visualize the results, some examples are shown in Figure 2, which contains graphs generated by our model and the baselines. Although it is not as intuitive for humans to assess as, e.g., natural images, one can still see that LGGAN appears to capture the typical structures of datasets better than other models.

Scalability To evaluate the scalability of these methods, we perform experiments on two different subsets of the Cora dataset with different graph sizes: the cora_small and cora datasets. As listed in Table 1, the traditional models all create a large gap between these two datasets in terms of three evaluation metrics. For the deep generative models, DeepGMG cannot be used to generate large graphs due to the computational complexity of it's generation procedure which try to add node one by one increasingly. And compared to GraphRNN, LGGAN MMD scores barely increase compare to the smaller dataset, suggesting that our model is more reliable and has the best ability to scale up to large graphs.

Generality To evaluate the ability of LGGAN to adapt to different graph-structured data, we evaluate the results of all methods on the different domains of citation ego-networks (Cora) and molecular protein graphs (ENZYMES). From Table 1, LGGAN achieves more consistent results on various datasets compared to other models, where some of them suffer from the issue of generalization such as MolGAN and NetGAN which can only be used to generate specific or limited types of graph-structured data.

4 Conclusion

In this work, we proposed a deep generative model using a GAN framework that generates labeled graphs. These labeled graphs can mimic distributions of citation graphs, knowledge graphs, social networks, and more. Our model can be useful for simulation studies, especially when access to labeled graph data is limited by access or privacy concerns. We can use these models to generate synthetic datasets or augment existing datasets, to do graph-based analyses such as communication segmentation, node classification, anomaly detection, and link prediction. The experiments show that it outperforms other state-of-the-art models for generating graphs while also being capable of the previously unaddressed task of generating labels for nodes.

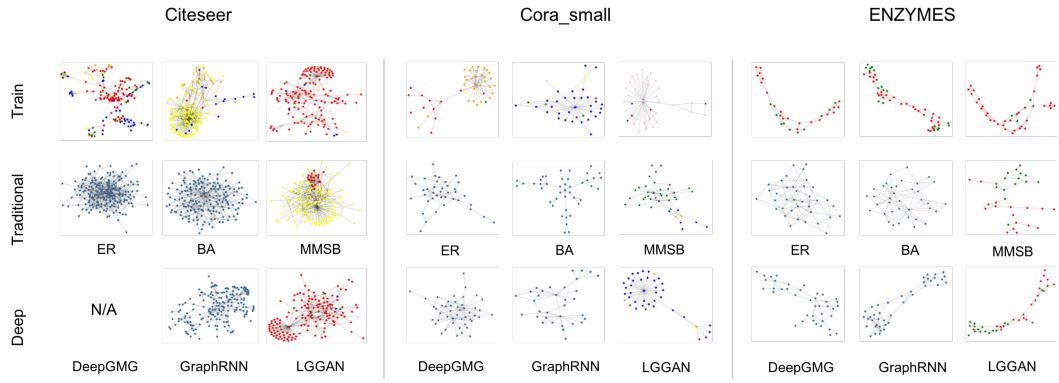

Figure 2: Visualization of training graphs (first row), graphs generated by traditional models (second row): E-R model, B-A model, MMSB model and graphs generated by deep models (third row): DeepGMG, GraphRNN, LGGAN for different datasets.

REFERENCES

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed Membership Stochastic Blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

Réka Albert and Albert-László Barabási. Statistical Mechanics of Complex Networks. *Reviews of modern physics*, 74(1):47, 2002.

Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. NetGAN: Generating graphs via random walks. In *International Conference on Machine Learning*, 2018.

Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Paul D Dobson and Andrew J Doig. Distinguishing Enzyme Structures from Non-enzymes Without Alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

P. Erdös and A. Rényi. On Random Graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651. JMLR. org, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, 2018.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pp. 5694–5703, 2018.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915, 2017.