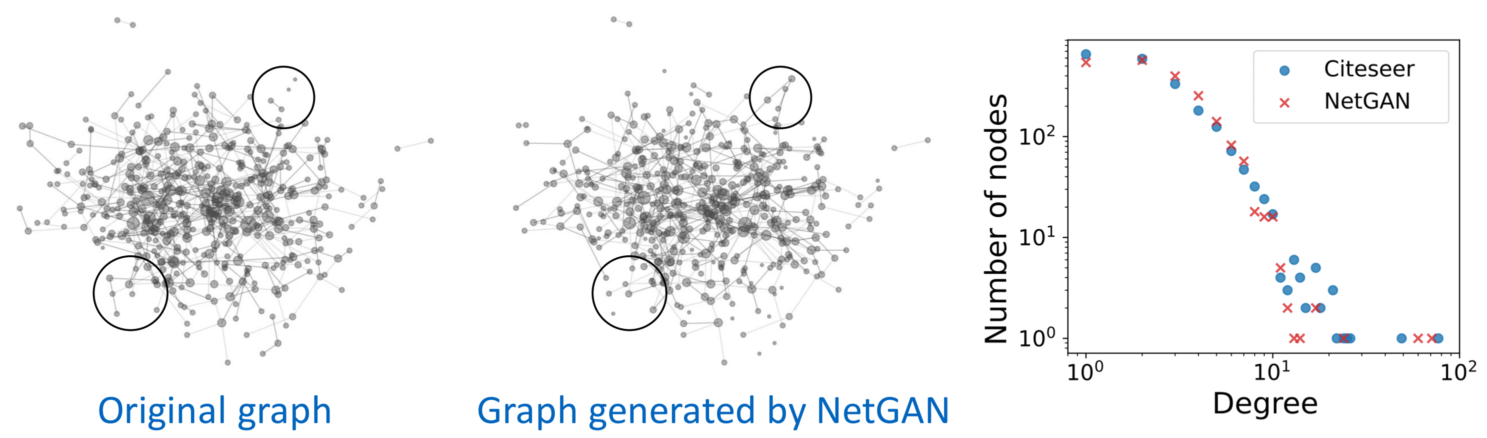
Professorship of Data Mining and Analytics
Department of Informatics
Technical University of Munich

NetGAN: Generating Graphs via Random Walks

Aleksandar Bojchevski*, Oleksandr Shchur*, Daniel Zügner*, Stephan Günnemann
* equal contribution

Motivation

- ❑ Complex networks are everywhere: social networks, protein interaction networks, e-commerce systems, the Web, …
- ❑ Generative models are useful for analyzing complex networks
- ❑ Applications of generative models: e.g. imputation of missing data, simulation, controllable generation of new data



Original graph   Graph generated by NetGAN

Generative Models for Graphs

- ❑ Structure of real-world networks follows common patterns
- ❑ Prescribed models directly specify these properties in the model definition

	Sparsity	Power law degree dist.	Community structure	Triangle counts	Homophily	Undiscovered patterns
Planted Part.	✓	✗	✓	✗	✓	✗
DC-SBM	✓	✓	✓	✗	✓	✗
ERGM	✓	✓	✗	✓	✓	✗
NetGAN	✓	✓	✓	✓	✓	(✓)

- ❑ Can we design a model that captures all the patterns without specifying them?
- ❑ What about patterns we haven't found yet? What if our current beliefs are wrong?

Let the data speak for itself!
Our goal: Design a deep implicit model for graphs

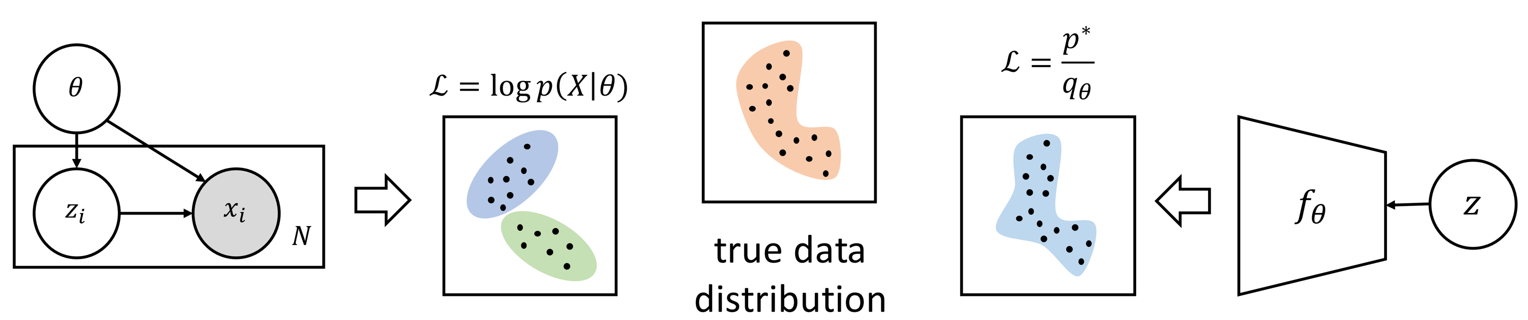
Challenges

- ❑ We want to learn from a single large graph, thus we have only one training example
- ❑ Scalability: N^2 edges are possible in a graph with N nodes, but graphs are sparse
- ❑ Graphs are discrete (unlike e.g. images, audio)
- ❑ Graphs are invariant under node reordering

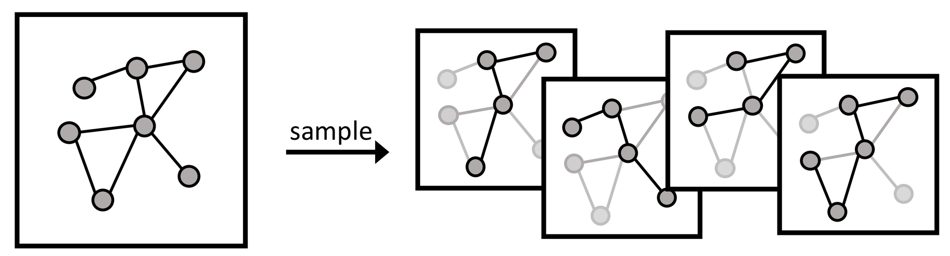
Background: Implicit Generative Models

- ❑ Prescribed models provide an explicit parametric specification of a distribution
- ❑ Implicit models models define a stochastic process that directly generates data

Prescribed model   Implicit model



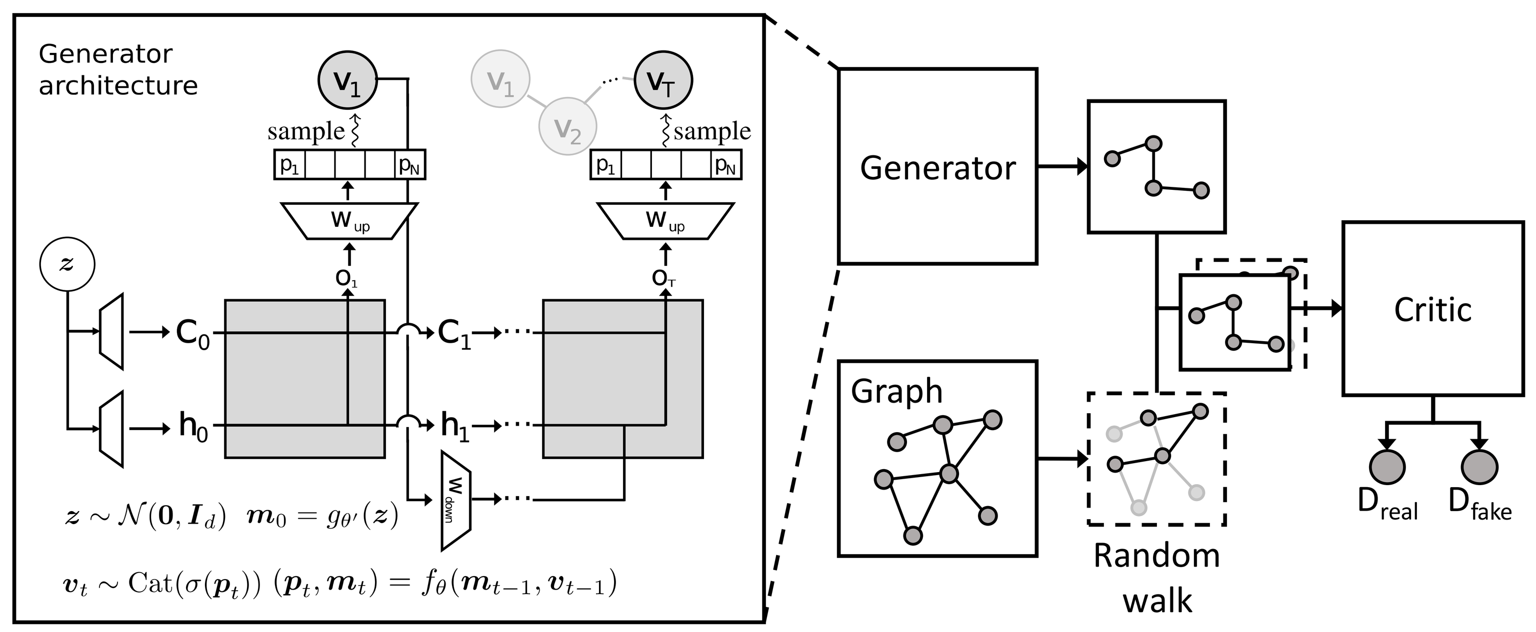
NetGAN: Idea



Learning to generate graphs by learning a distribution over random walks
- ☑ Exploits sparsity in networks
- ☑ Transforms a single large network into many training sequences

NetGAN: Model

Deep implicit generative model for graphs trained using the Wasserstein GAN principle



Generator

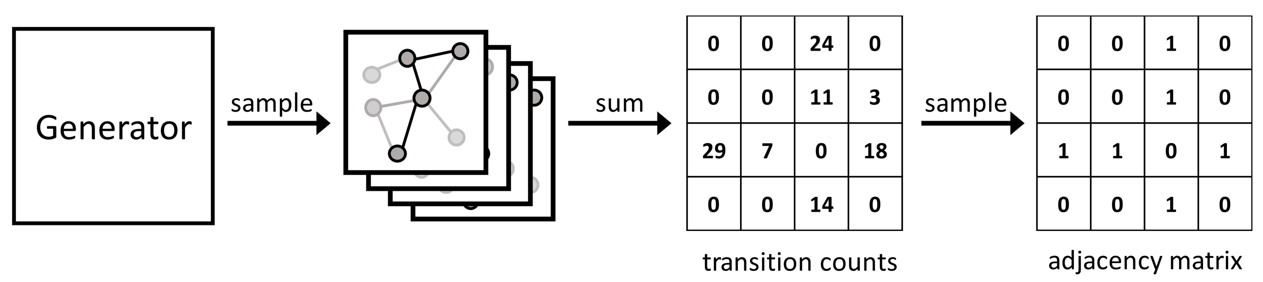
- ❑ Stochastic recurrent neural network that generates sequences of discrete samples corresponding to random walks
- ❑ Sample each node in the sequence conditioned on the previous node and the memory
- ❑ LSTM operates in a low dimensional space $H \ll N$ for efficiency

Training

- ❑ The generator is trained using the Wasserstein GAN principle
- ❑ The critic is an LSTM that outputs a plausibility score given a random walk
- ❑ Tackling discreteness: Gumbel-Straight-Through estimator (Jang et al. 2017) enables backpropagation through non-differentiable sampling from a categorical distribution

Assembling the adjacency matrix

Each edge is sampled with probability proportional to the number of times it appears in the random walks generated by NetGAN



Experimental Evaluation

NetGAN is able to capture graph patterns …

	Max degree	Power law exp.	Intra-Com. density	Inter-Com. density	Clustering coefficient	Character. path len.
Original (Cora-ML)	240	1.86	1.7e-3	4.3e-4	2.7e-3	5.61
Configuration model	240	1.86	2.8e-4	1.6e-3	3.0e-4	4.38
DC-SBM	165	1.81	1.2e-3	6.7e-4	3.3e-3	5.12
ERGM	243	1.79	1.2e-3	6.9e-4	2.2e-3	4.59
BTER	199	1.79	7.5e-4	1.0e-3	4.6e-3	4.59
VGAE	13	1.67	3.2e-4	1.4e-3	1.2e-3	5.28
NetGAN	233	1.79	1.4e-3	6.0e-4	2.4e-3	5.20

… better than prescribed models without manually specifying them

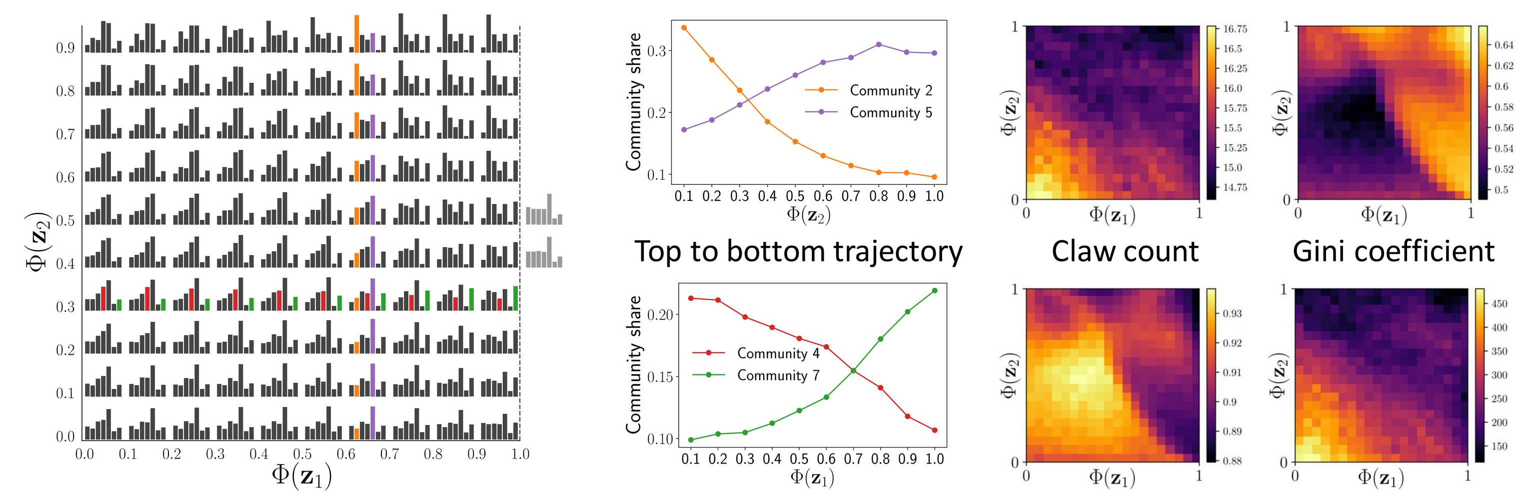
NetGAN generalizes

We use the random walk transition counts generated by NetGAN to evaluate its generalization via link prediction

	Cora-ML	Citeseer	Pubmed	PolBlogs	DBLP	Cora
Number of nodes	2.8 K	2.1 K	19.7 K	1.8 K	16.2 K	18.8 K
Number of edges	7.9 K	3.7 K	44.3 K	16.7 K	51.9 K	64.5 K
Adamic/Adar	92.16	88.69	84.98	85.43	91.13	93.00
DC-SBM	**96.03**	94.77	**96.76**	95.46	**97.05**	98.01
node2vec	92.19	95.29	96.49	85.10	96.41	**98.52**
VGAE	95.79	95.11	94.50	93.73	96.38	97.59
NetGAN (500K)	94.00	95.18	87.39	95.06	82.45	82.31
NetGAN (100M)	95.19	**96.30**	93.41	**95.51**	86.61	84.82

- ❑ NetGAN achieves competitive results on small and medium sized graphs
- ❑ On large datasets more random walks are needed to get representative transition counts

Latent space interpolation



Community histograms   Top to bottom trajectory   Left to right trajectory   Claw count   Gini coefficient   Rel. edge dist. entropy   Maximum degree

Interpolation in the latent space produces graphs with smoothly changing properties

Limitations and Future Work

- ❑ Generate graphs with varying number of nodes
- ❑ Scaling up to massive networks
- ❑ Evaluation metrics for graph generative models
  - ▪ Quality of generated graphs is hard to (visually) assess
  - ▪ New metrics can improve our understanding of graph generative models