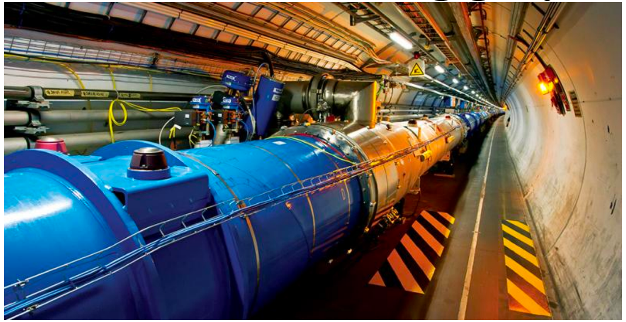
Graphite: Iterative Generative Modeling of Graphs

Aditya Grover
Stanford University

Joint work with Aaron Zweig and Stefano Ermon

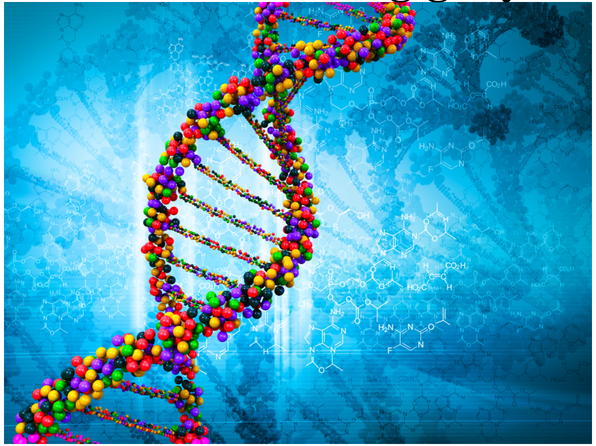
Age of big **unlabeled** data

25 million gigabytes



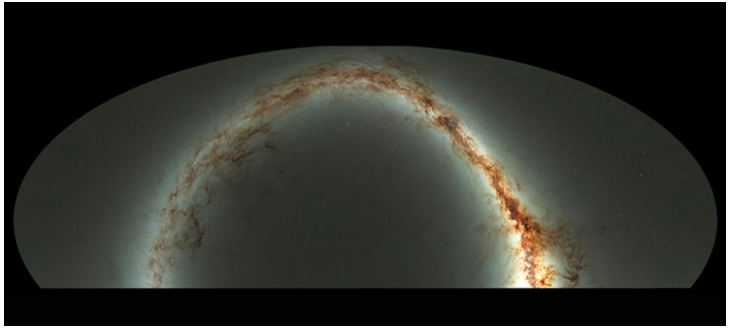
Large Hadron Collider

15 million gigabytes



European Bioinformatics Institute

2 million gigabytes

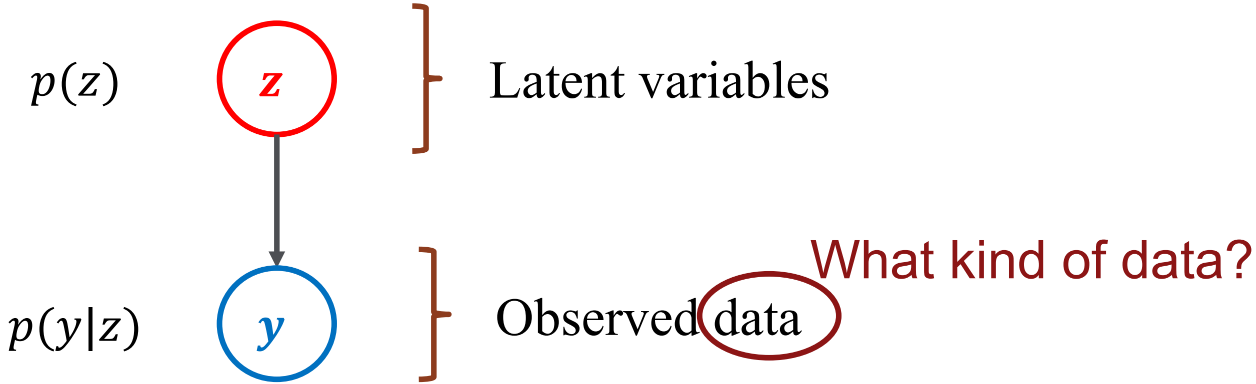


Pan-STARRS database

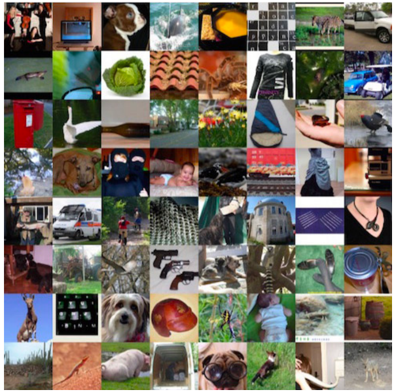
How do we make **inferences** over **unlabeled data**?

Generative modeling

- Learns a probability distribution over data a.k.a. **density estimation**
- Provides a simulator for the data a.k.a. **sampling**
- Learns latent features for the data a.k.a. **representational learning**

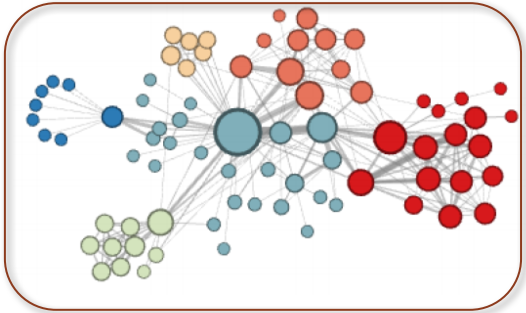


Different **modalities** of structured data



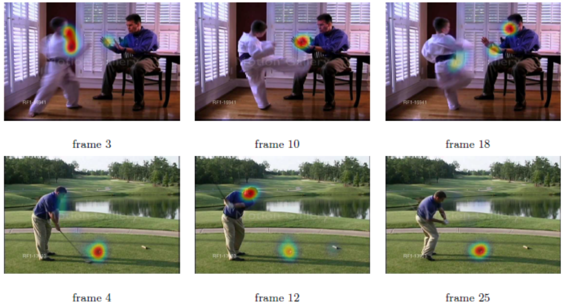
Images

Audio



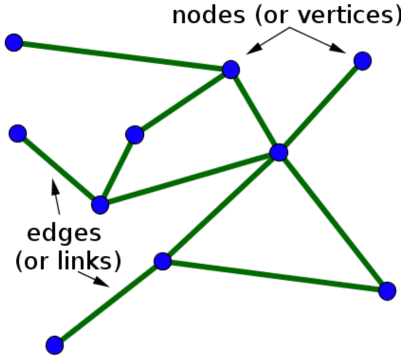
Graphs

Text



Video

Graphs are **ubiquitous**



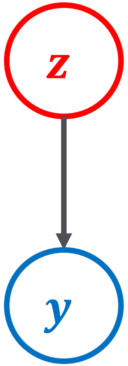
$$G = (\mathbf{A}, \mathbf{X})$$

Adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$
Feature matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$

Ecology: Food web networks
Biology: Brain networks, Protein-protein interaction networks
Chemistry: Molecules, materials
…

Learning **deep latent variable models** of graphs

$p(z)$

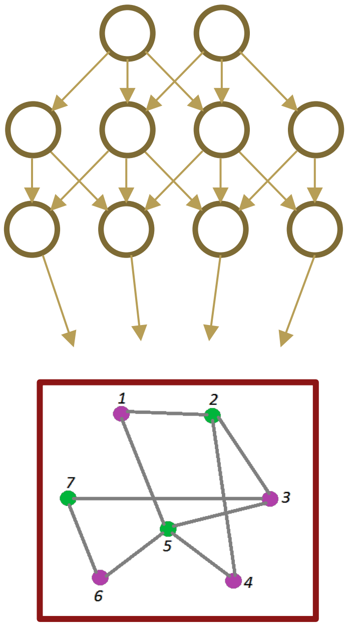


$p(y|z)$

$p(\mathbf{Z})$ $\quad \mathbf{Z} \in \mathbb{R}^{n \times k}$

H_1

H_2



$p_\theta(\mathbf{A}|\mathbf{X}, \mathbf{Z})$ $\quad G = (\mathbf{A}, \mathbf{X})$

What is the right **network architecture** for graphs?

Images – Spatial structure – Convolutional Neural Networks (CNN)

Text, Speech – Temporal structure – Recurrent Neural Networks (RNN)

Video – Spatiotemporal structure – Hybrids of CNNs and RNNs

Inductive biases and invariances for graphs?

- Local structure in terms of graph neighborhoods
- Permutation invariance to node reorderings
- Dynamic resizing

Graph Convolutional Networks (Kipf and Welling, 2017)

Graph Convolutional Networks

- A spectral graph convolution is defined as the multiplication of a signal (i.e., **X**) with a parameterized filter F_θ in the Fourier space of a graph:

$$F_\theta * \mathbf{X} = \mathbf{U} F_\theta \mathbf{U}^T \mathbf{X}$$

with **U** as the left eigenvector matrix of the graph Laplacian.

- **Graph convolutional networks** compute an efficient first order approximation. Forward pass from $\mathbf{H}^{(l-1)}$ to $\mathbf{H}^{(l)}$:

$$\mathbf{H}^{(l)} = \eta(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{H}^{(l-1)}\mathbf{\Theta}^{(l)})$$

with non-linearity η, degree matrix $\mathbf{D}$, and parameters $\theta^{(l)}$.

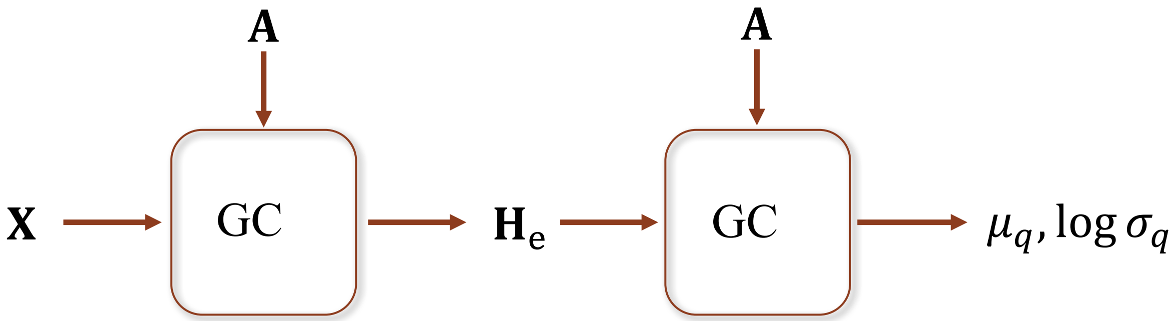
Variational Autoencoding using **Graphite**

- Maximizing the marginal log-likelihood $\log p_\theta(\mathbf{A}|\mathbf{X})$ is intractable
- Introduce a variational posterior $q_\phi(\mathbf{Z}|\mathbf{A},\mathbf{X})$ parameterized by ϕ
- Maximize an evidence lower bound (ELBO) to the log-likelihood

$$\log p_\theta(\mathbf{A}|\mathbf{X}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{A},\mathbf{X})}\left[\log \frac{p_\theta(\mathbf{A},\mathbf{Z}|\mathbf{X})}{q_\phi(\mathbf{Z}|\mathbf{A},\mathbf{X})}\right]}_{ELBO(\theta,\phi)}$$

Graphite **Encoder**

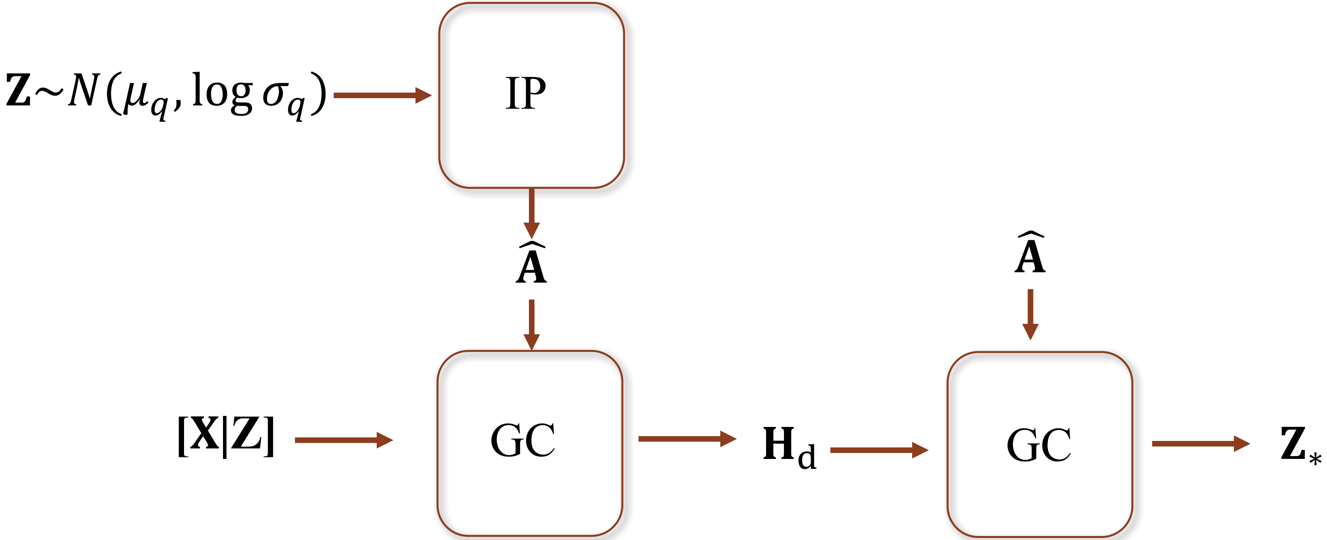
- Variational posterior $q_\phi(\mathbf{Z}|\mathbf{A},\mathbf{X})$ is a multivariate Gaussian with diagonal covariance
- Encoder parameterized by a graph convolutional network



Forward pass of a two layer encoding GCN

Graphite **Decoder**

- Decoder is a hybrid that iterates between:
  - intermediate graph construction using an inner product decoder
    - $\mathrm{IP}(\mathbf{Z}) = \mathrm{sgm}(\mathbf{Z}\,\mathbf{Z}^{\mathrm{T}})$
  - message passing on the intermediate graph using graph convolutions



Graphite **Decoder**

- The final latent feature matrix is specified as a convex combination of the latent layers

$$\mathbf{Z}' = \lambda \mathbf{Z} + (1-\lambda)\mathbf{Z}_*$$

where $\lambda \in [0,1]$ is a tunable hyperparameter.

- Observation model $p_\theta(\mathbf{A}|\mathbf{X},\mathbf{Z})$ is a factorized multivariate Bernoulli

$$p_\theta(\mathbf{A}|\mathbf{Z},\mathbf{X}) = \Pi_{i=1}^{n}\Pi_{j=1}^{n} p_\theta(A_{ij}|\mathbf{Z},\mathbf{X})$$
$$\text{where } p_\theta(A_{ij}|\mathbf{Z},\mathbf{X}) = \sigma(\mathbf{Z}_i'\mathbf{Z}_j')$$

Link Prediction

- Given two nodes in a graph, does an edge exist between the nodes?
- **Baselines:**
  - Spectral Clustering (SC)
  - DeepWalk (DW): random walks + skipgram objective
  - (Variational) Graph Autoencoder (VGAE, GAE): GCN encoder but a single-step inner product decoder
- For Graphite, the task can be formulated as denoising.
- **Datasets:** Protein-protein Interaction, Cora, Citeseer, Pubmed
- **Evaluation metrics:** Area Under the ROC Curve and Average Precision

Evaluation for Link Prediction

Table 1: Area Under the ROC Curve (AUC) scores for link prediction

	PPI	Cora	Citeseer	Pubmed
SC	84.2 ± 0.34	89.9 ± 0.20	91.5 ± 0.17	$\mathbf{94.9} \pm 0.04$
DW	68.2 ± 0.08	85.0 ± 0.17	88.6 ± 0.15	91.5 ± 0.04
GAE	88.8 ± 0.01	90.2 ± 0.16	92.0 ± 0.14	92.5 ± 0.06
VGAE	89.5 ± 0.07	90.1 ± 0.15	92.0 ± 0.17	92.3 ± 0.06
Graphite-AE	91.1 ± 0.05	$\mathbf{91.4} \pm 0.16$	92.5 ± 0.16	94.5 ± 0.05
Graphite-VAE	$\mathbf{91.2} \pm 0.05$	$\mathbf{91.4} \pm 0.16$	$\mathbf{93.0} \pm 0.12$	94.6 ± 0.04

State-of-the-art on **link prediction**.

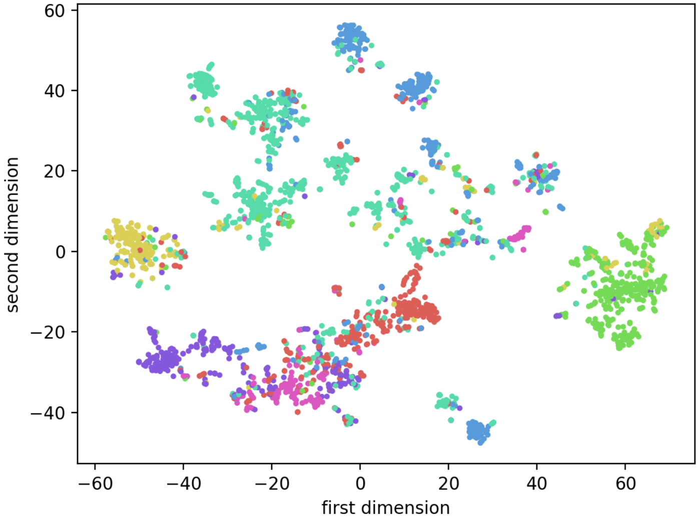
Evaluation for Link Prediction

Table 2: Average Precision (AP) scores for link prediction

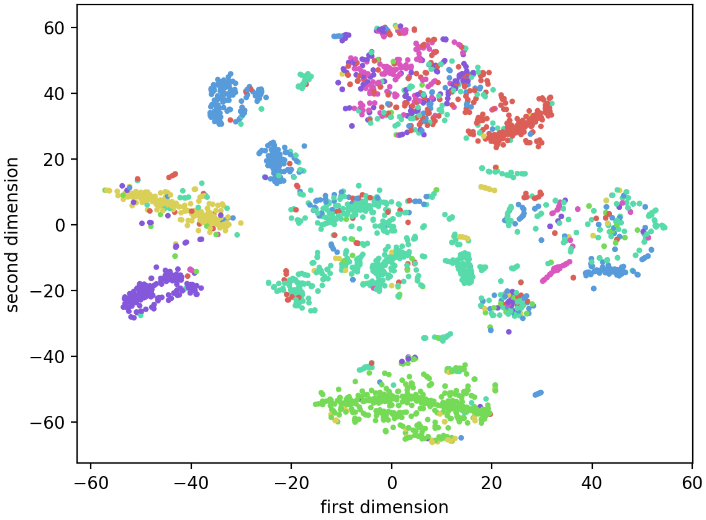
	PPI	Cora	Citeseer	Pubmed
SC	88.9 ± 0.21	92.8 ± 0.12	94.4 ± 0.11	$\mathbf{96.0} \pm 0.03$
DW	69.0 ± 0.09	86.6 ± 0.17	90.3 ± 0.12	91.9 ± 0.05
GAE	89.4 ± 0.05	92.4 ± 0.12	94.0 ± 0.12	94.3 ± 0.5
VGAE	89.6 ± 0.05	92.3 ± 0.12	94.2 ± 0.12	94.2 ± 0.04
Graphite-AE	92.1 ± 0.05	92.4 ± 0.17	93.5 ± 0.19	95.7 ± 0.06
Graphite-VAE	$\mathbf{92.2} \pm 0.06$	$\mathbf{93.1} \pm 0.13$	$\mathbf{94.6} \pm 0.12$	$\mathbf{96.0} \pm 0.03$

Graphite outperforms **competing methods** on both ROC and AP metrics!

Visualization of Latent Space



Graphite Autoencoder



Graphite Variational Autoencoder

Cora Dataset

Conclusion

- Proposed **Graphite**, an algorithmic framework for generative modeling of graphs using variational autoencoding.
- Outperforms state-of-the-art methods for **link prediction**.
- Future and ongoing work entails applications of Graphite to other inference tasks such as **graph synthesis** and **semi-supervised node and graph classification**.