SGVAE: Sequential Graph Variational Autoencoder

Bowen Jing, Ethan A. Chi, Jillian Tang

{bjing, ethanchi, jiltang}@cs.stanford.edu

Background

- Graphs are a natural way to represent information in many domains, such as social networks or molecules.
- Existing graph generative models are limited in scalability and expressivity [1].
- Li et al. present a step-by-step, autoregressive generative model that performs well but has low interpretability [2].
- We develop SGVAE (a sequential graph variational autoencoder) to extend this model to learn a latent space.

Model

- We seek to model

$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$$

  where $p_\theta(\mathbf{x}|\mathbf{z})$ is given by Li et al's sequential generative method [2] and $p_\theta(\mathbf{z})$ is a latent variable prior.
- Our latent prior takes the form of an initial embedding $\mathbf{z}$ and an ordering $\pi \in \Pi$ of the nodes of $\mathbf{x}$.
- We introduce an amortized variational model over $\mathbf{z}, \pi$:

$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z},\pi\sim q_\phi(\mathbf{z},\pi|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\pi|\mathbf{z})}{q_\phi(\mathbf{z},\pi|\mathbf{x})}p_\theta(\mathbf{z})\right]$$

- We parametrize $p_\theta(\mathbf{x},\ \pi|\mathbf{z})$ by a **constructor** network [2] initialized to $\mathbf{z}$:

$$\mathbf{h}_V^{(T)} = \text{prop}^{(T)}(\mathbf{h}_V, G)$$
$$\mathbf{h}_G = R(\mathbf{h}_V^{(T)}, G)$$
$$f_{addnode}(G) = \text{softmax}(f_{an}(\mathbf{h}_G))$$
$$f_{addedge}(G, v) = \sigma(f_{ae}(\mathbf{h}_G, \mathbf{h}_v^{(T)}))$$
$$s_u = f_s(\mathbf{h}_u^{(T)}, \mathbf{h}_v^{(T)}), \quad \forall u \in V$$
$$f_{nodes}(G, v) = \text{softmax}(\mathbf{s})$$

- We parametrize $q_\phi(\mathbf{z}, \pi|\mathbf{x})$ by a novel **destructor** network:

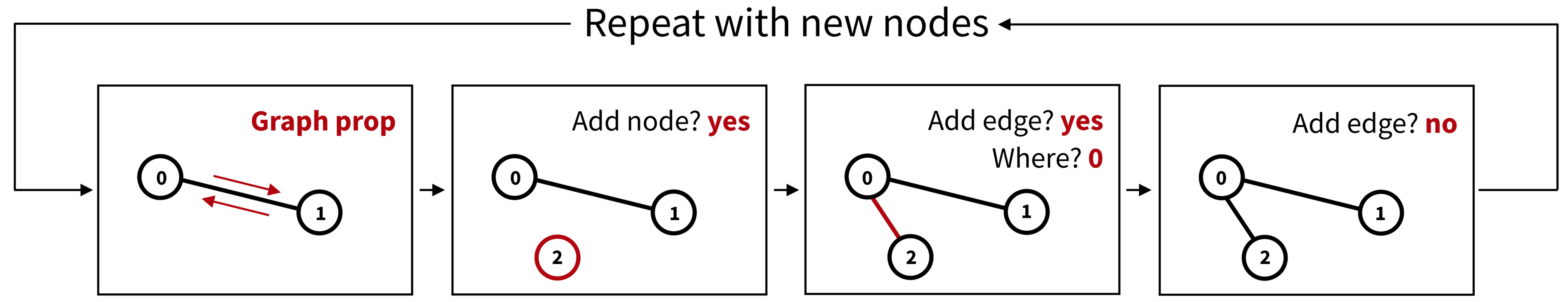
$$\mathbf{h}_V^{(T)} = \text{prop}^{(T)}(\mathbf{h}_V, G)$$
$$\mathbf{h}_G = R(\mathbf{h}_V^{(T)}, G)$$
$$f_{remove}(G) = \text{softmax}(f_{an}(\mathbf{h}_G))$$

- Since π is discrete, cannot use the reparametrization trick to estimate $\nabla_\phi q_\phi(\mathbf{z}, \pi|\mathbf{x})$. Instead, use the REINFORCE trick:

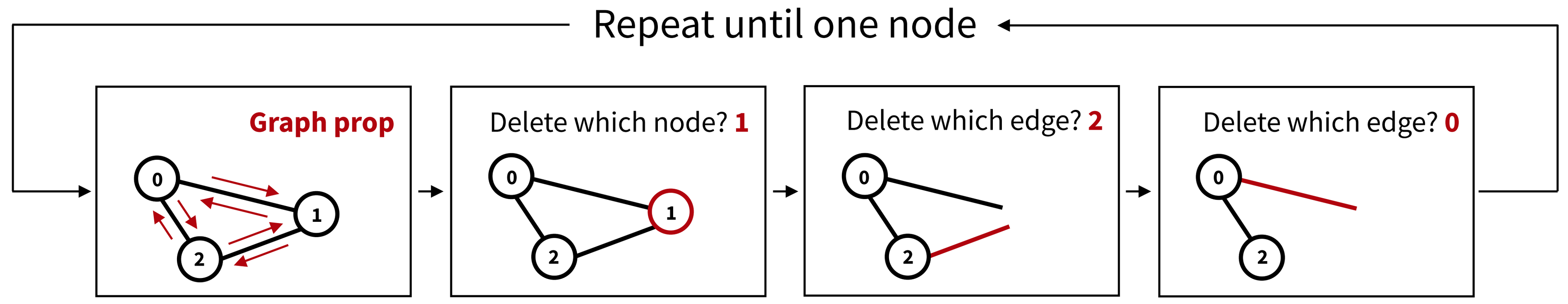
$$\frac{1}{n}\sum_{i=1}^{n}\left[\nabla_\phi[\log q_\phi(\mathbf{z}^{(i)},\pi^{(i)}|\mathbf{x})]\left[\log\left(\frac{p_\theta(\mathbf{x},\pi^{(i)}|\mathbf{z}^{(i)})}{q_\phi(\mathbf{z}^{(i)},\pi^{(i)}|\mathbf{x})}p_\theta(\mathbf{z}^{(i)})\right)-1\right]\right]$$

Graph Processing

Construction

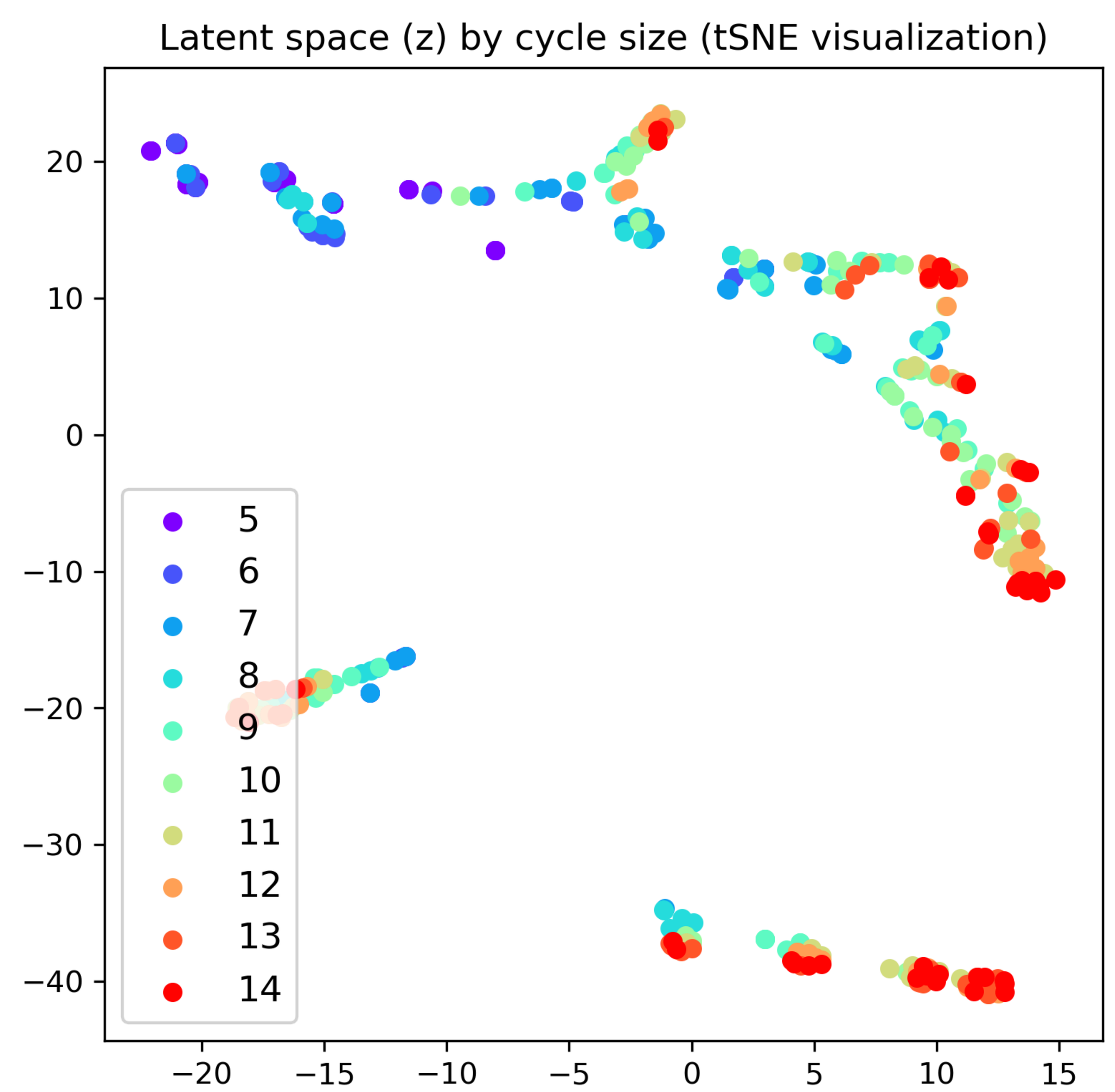


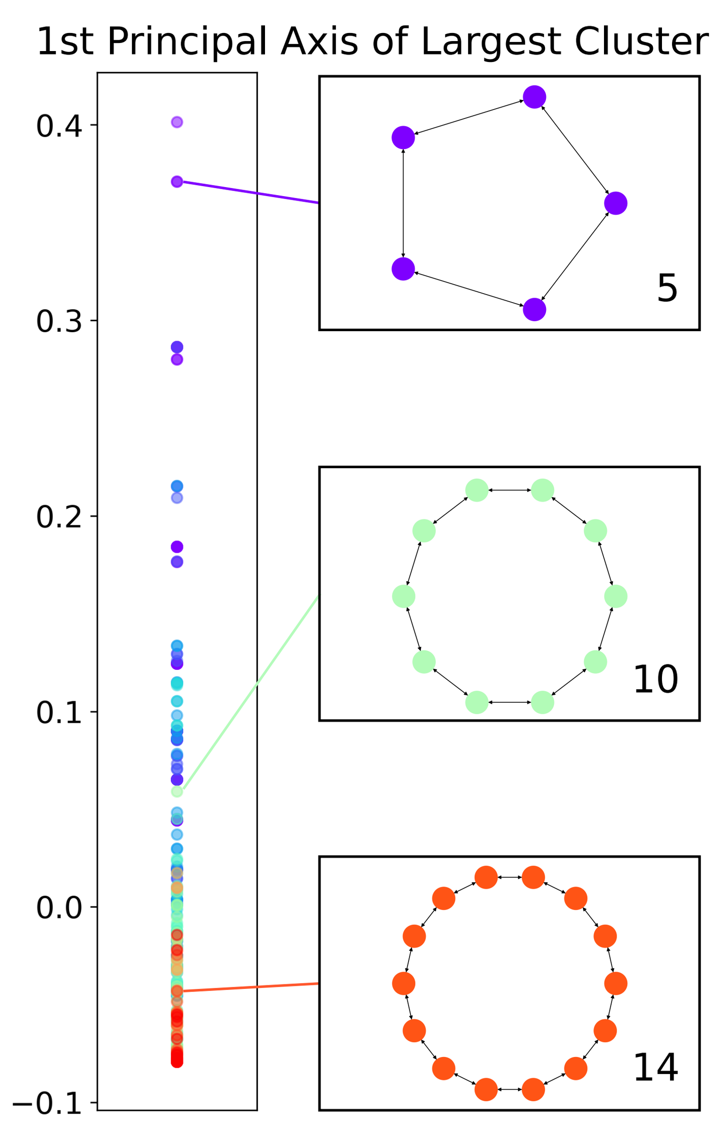
Destruction



Cycles: Latent Space

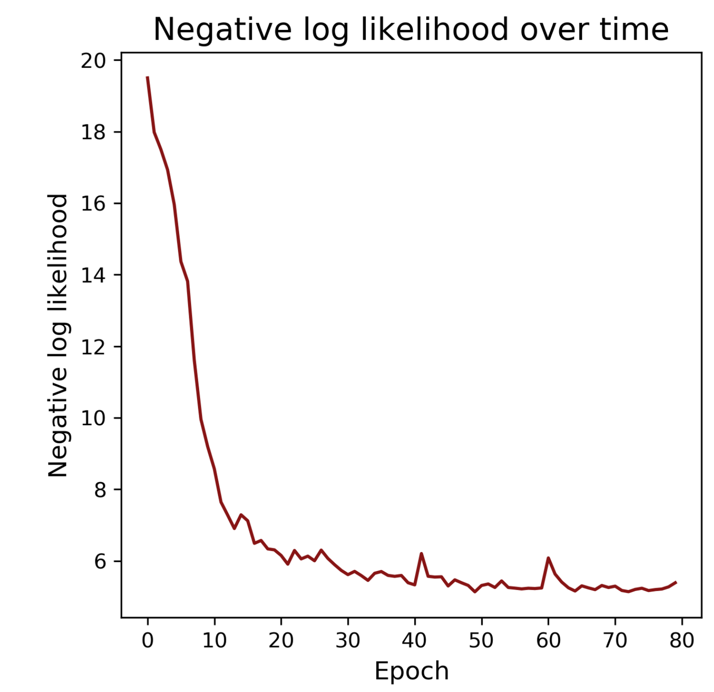
- We find that SGVAE learns an interpretable, meaningful latent space.
- We visualize the 5-dimensional latent space using tSNE and find that it separates naturally into several clusters, each of which is approximately linear in cycle size as visualized by PCA.

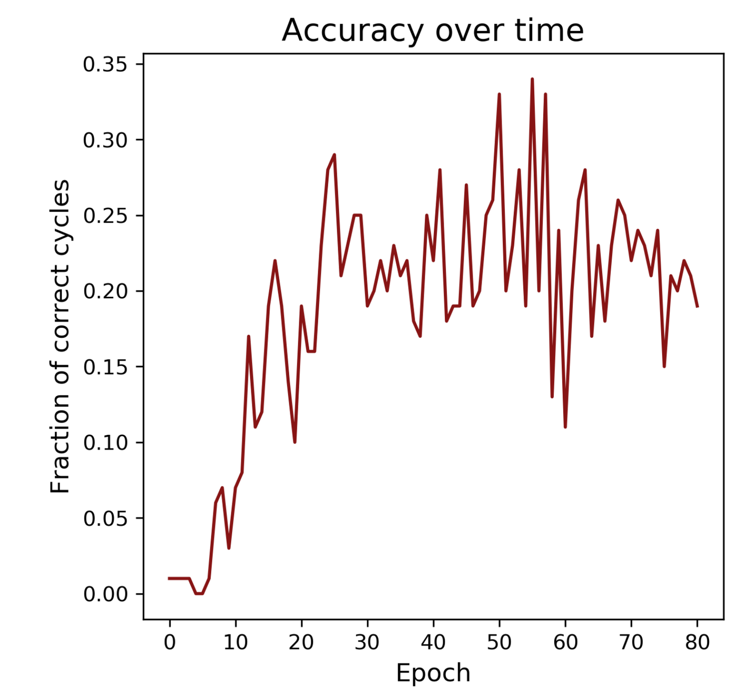




Cycles: Experiments

- As a first toy experiment, we train on cycles of 5-15 nodes.
- We achieve lower perplexity on the cycles dataset than the original Li et al. model.





Analysis

- We find that the destructor network is able to learn an interpretable, meaningful latent space.
- Since π is discrete, it is difficult to learn a canonical ordering. This makes the latent space of $\mathbf{z}$ sparse.
- Although each cluster is linearly correlated with attributes of the input graphs, there is no cross-cluster correlation.
- Because this latent space is so sparse, sampling is difficult (but dimensionality reduction may lead to good sampling).
- **Future work:**
  - Improving training using control variates
  - Better generation through beam search

Citations

[1] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*.

[2] Li, Y., Vinyals, O., Dyer, C., Pascanu, R., & Battaglia, P. (2018). Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*.