Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders

{ Tengfei Ma*, Jie Chen*, Cao Xiao } IBM Research

Introduction

In this work, we propose a regularization framework for VAEs to generate **semantically valid** graphs. Contributions:

1. A new deep generative framework for general graph generation with validity constraints.
2. Formulation of constraints for **I. molecular graphs** and **II. node-compatible graphs**.
3. An efficient inference algorithm for generating valid graphs.

Examples

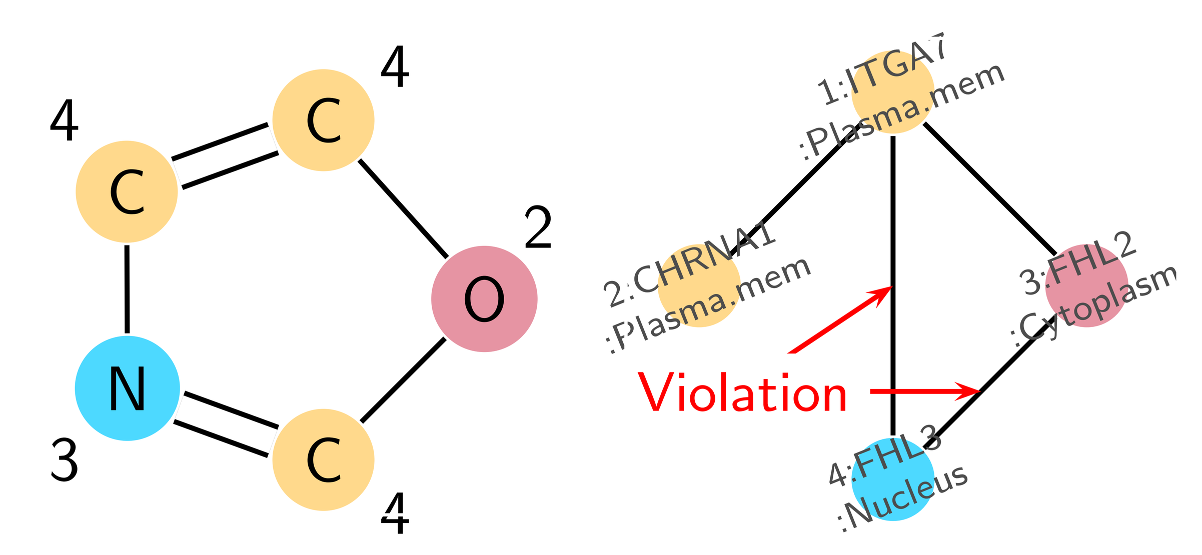

Figure 1: Left: Moleculer graph. Right: Protein network.

Framework

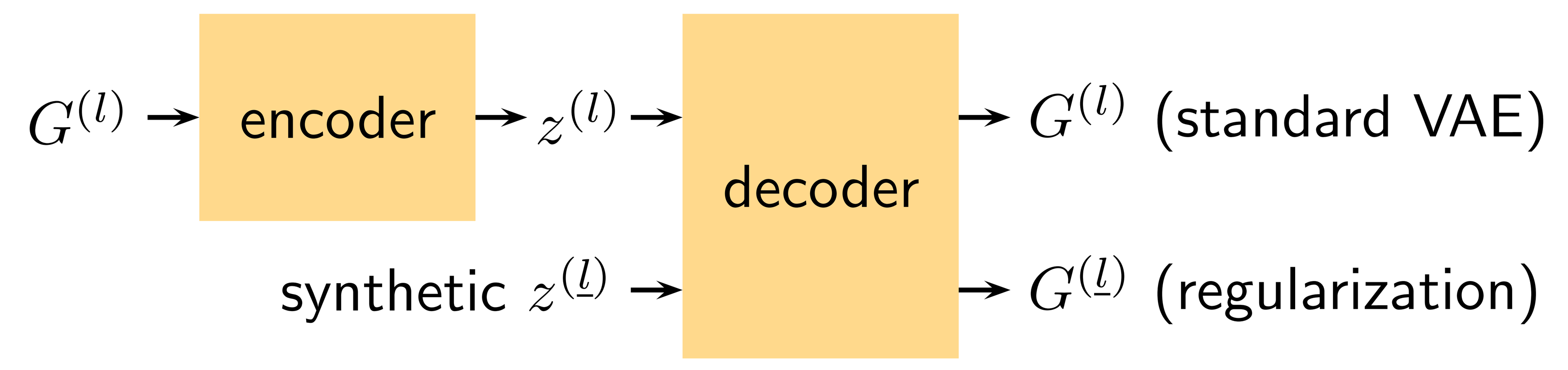

Figure 2: Overview of the Regularized VAE

The central contribution of this work is an approach to imposing validity constraints in the training of VAEs. Besides the usual loss function $f(x)$ of VAE, we want the graph samples produced by the generative network to be valid, regardless of what latent value z one starts with:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & \text{for almost all } z \sim p_x(z), \\ & h_1(x,z)=0,\dots,h_m(x,z)=0, \\ & g_1(x,z)\le 0,\dots,g_r(x,z)\le 0. \end{aligned} \tag{1}$$

Lagrangian: To solve (1), we generalize the usual notion of Lagrangian function to

$$\mathcal{L}(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i \widetilde{h}_i(x) + \sum_{j=1}^{r} \mu_j \widetilde{g}_j(x), \tag{2}$$

where $\widetilde{g}_j(x) = \left[\int g_j(x,z)^2 p_x(z)\,dz\right]^{\frac{1}{2}}$ and similarly for $\widetilde{h}_i(x)$.

Training: Let us write the i-th validity constraint as $g_i(\theta,z)\le 0$ for all z, then the *loss function* is

$$-L_{\text{ELBO}}(\theta,\phi) + \mu \sum_i \left[\int g_i(\theta,z)_+^2 p_\theta(z)\,dz\right]^{\frac{1}{2}}, \tag{3}$$

where $g_+ = \max(g,0)$ is the ramp function. This regularization will not penalize the desirable case $g_i \le 0$.

In practice, the integral in the regularization may be intractable, and hence we appeal to *Monte Carlo approximation* for evaluating the loss in each parameter update:

$$-L_{\text{ELBO}}(\theta,\phi) + \mu \sum_i g_i(\theta,z)_+, \quad \text{where} \quad z \sim p_\theta(z). \tag{4}$$

Visualization

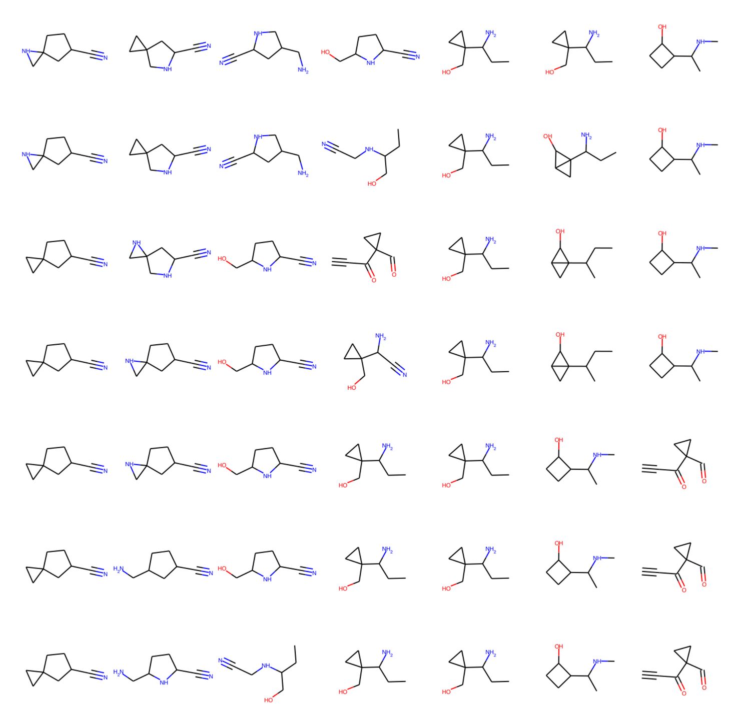

Figure 3: 2D interpolation of latent space.

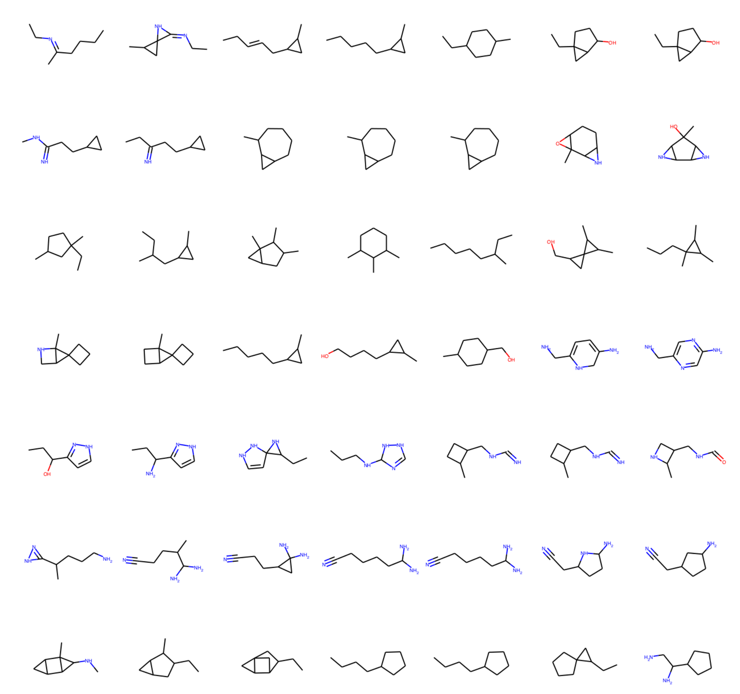

Figure 4: 1D interpolation of latent space (each row).

Results

Table 1: Standard VAE versus regularized VAE.

Method	% Valid	ELBO
QM9		
Standard	83.2	-17.3
Regul.	**96.6**	**-18.5**
Node-compatible		
Standard	40.2	-42.5
Regul.	**98.4**	**-51.2**

Table 2: Comparison with other VAEs.

Method	% Valid	% Novel	% Recon.
QM9			
Proposed	**96.6**	**97.5**	**61.8**
GVAE	60.2	80.9	96.0
CVAE	10.3	90.0	3.61
ZINC			
Proposed	**34.9**	**100**	**54.7**
GVAE	7.2	100	53.7
CVAE	0.7	100	44.6

Table 3: Denoising node-incompatible graphs.

Standard VAE	Regularized VAE
11.2	**93.8**

Regularization

Molecules:

- **Ghost Nodes and Valence.** Denote by $V(i)$ the capacity of a node i and by $U(i)$ the valence (upper bound of the capacity). The valence constraint is $g_i = V(i) - U(i)$.
- **Connectivity**: If A is the adjacency matrix, then the (i,j) element of $B = I + A + A^2 + \cdots + A^{N-1}$ is nonzero iff i and j are connected by a path. Let $q(i)=0$ indicate the ghost node, then the connectivity constraint is $q(i)q(j)\cdot \mathbf{1}\{B(i,j)=0\} + [1-q(i)q(j)]\cdot \mathbf{1}\{B(i,j)\neq 0\} \le 0$. To be differentiable, in our framework $g_{ij} = q(i)q(j)\cdot[1-C(i,j)] + [1-q(i)q(j)]\cdot C(i,j)$ where $C=\sigma(B)$.

Node-compatible Graphs: Consider the matrix $P = \widetilde{F} D \widetilde{F}^T$. The (i,j) element of P is the probability that nodes i and j have compatible types. We want node pairs with low compatibility to be disconnected. Hence, the constraint is $g_{ij} = [1-\widetilde{E}(i,j,0)][1-P(i,j)] - \alpha$ where $\alpha \in (0,1)$ is a tunable hyperparameter.

Network Architecture

For input representation, we unfold the edge-label tensor $E \in \mathbb{R}^{N\times N\times(1+t)}$ and concatenate it with the node-label matrix $F \in \mathbb{R}^{N\times(1+d)}$ to form a wide matrix. The encoder is a 4-layer convolutional neural net. The decoder is a 4-layer deconvolutional neural net similar to DCGAN.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[2] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *ICML*, 2017.