Learning Latent Space Energy-Based Prior Model for Molecule Generation

Bo Pang[1] **Tian Han**[2] **Ying Nian Wu**[1]
[1]University of California, Los Angeles [2]Stevens Institute of Technology
bopang@ucla.edu than6@stevens.edu ywu@stat.ucla.edu

Abstract

Deep generative models have recently been applied to molecule design. If the molecules are encoded in linear SMILES strings, modeling becomes convenient. However, models relying on string representations tend to generate invalid samples and duplicates. Prior work addressed these issues by building models on chemically-valid fragments or explicitly enforcing chemical rules in the generation process. We argue that an expressive model is sufficient to implicitly and automatically learn the complicated chemical rules from the data, even if molecules are encoded in simple character-level SMILES strings. We propose to learn latent space energy-based prior model with SMILES representation for molecule modeling. Our experiments show that our method is able to generate molecules with validity and uniqueness competitive with state-of-the-art models. Interestingly, generated molecules have structural and chemical features whose distributions almost perfectly match those of the real molecules.

1 Introduction

Designing molecules with desired properties is of vital importance in applications such as drug design and material science. Molecules are in the form of graphs. It is hence challenging to search for desirable ones in the molecule space. Recently, deep generative models have been applied to molecule modeling [2, 5, 11, 10, 8]. Most methods adopt Variational Autoencoder (VAE) model [4]. It embeds molecules into a continuous latent space, allowing for more efficient optimization, and then decodes the latent vector to a molecule, enabling new molecule generation.

In molecule modeling, two types of representations are widely used. One is simplified molecular input line entry systems (SMILES) [12] with which a molecule graph is linearized into a string consisting of characters that represent atoms and bonds. With this representation, an autoregressive model can be utilized to capture the chemical rules among atoms and bonds. The same model is widely used and called language model (LM) in natural language processing. Following [8], we call models adopting this representation as LM-based models. Another representation works directly with the graph where nodes and edges represent atoms and bonds respectively. Graph allows for explicitly encoding and directly enforcing chemical laws. To guarantee validity of generated molecules, many graph-based models [6, 9, 10] sequentially generate atoms (nodes) and bonds (edges), continuously check if the generated elements satisfy valency rules. Graph-based models are however more complicated and less efficient to train and sample from, compared to LM-based models.

Despite the simplicity and efficiency of LM-based models, they often produce invalid samples and duplicates. The recent work of [8] proposed FragmentVAE and argued that LM-based models can produce samples with perfect validity and uniqueness. Fragments are small-weight and chemically sound compounds, and FragmentVAE uses fragments instead of atoms as basic elements in molecule generation. To enhance uniqueness, FragmentVAE replaces infrequent fragments in generated molecules by new fragments that are uniformly sampled from a pool of infrequent fragments. These

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

techniques make the SMILES-fragment-based model competitive with the state-of-the-art graph-based models.

Instead of redesigning molecule representation or resorting to more complicated graph models, we argue that an expressive model is sufficient to capture the complicated chemical rules implicitly and generate valid and unique molecules, even with the character-level SMILES representation instead of fragment-level representation. Previous VAE-based methods rely on a generator network to map a prior distribution to be close to the data distribution and assume the prior to be a simple isotropic Gaussian distribution. Although a neural network generator is highly expressive, the assumption on the prior may cause ineffective learning of the model, which might explain why previous methods fail to generate valid and unique molecules without explicitly enforcing chemical rules. In this article, we propose to learn a latent space energy-based prior model [7] in addition to the generator network from observed molecules. Specifically, the prior model is an energy-based correction of the isotropic Gaussian distribution and the correction is learned from empirical data. Such a prior model improves the expressivity of the generator model. Our experiments demonstrate that our method is able to generate valid and unique samples, with the performance on par with the state-of-the-art models. Interestingly, we observe that the generated samples show structural and chemical properties (e.g., solubility, drug-likeness) that closely resemble the ground truth molecules.

2 Methods

2.1 Model

Let $x \in \mathbb{R}^D$ be an observed molecule such as represented in SMILES strings. Let $z \in \mathbb{R}^d$ be the latent variables, where $D \gg d$. Consider the following model,

$$z \sim p_\alpha(z), \quad x \sim p_\beta(x|z), \tag{1}$$

where $p_\alpha(z)$ is the prior model with parameters α, $p_\beta(x|z)$ is the top-down generative model with parameters β. In VAE, the prior is simply assumed to be an isotropic Gaussian distribution. In our model, $p_\alpha(z)$ is formulated as an energy-based model or a Gibbs distribution,

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z). \tag{2}$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian as in VAE. $f_\alpha(z)$ is the negative energy and is parameterized by a small multi-layer perceptron with parameters α. $Z(\alpha) = \int \exp(f_\alpha(z)) p_0(z) dz = \mathrm{E}_{p_0}[\exp(f_\alpha(z))]$ is the normalizing constant or partition function.

The generative model, $p_\beta(x|z)$, is a conditional autoregressive model,

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{3}$$

which is parameterized by a simple recurrent network with parameters β and $x^{(t)}$ indicates a one-hot encoded SMILES string.

It is worth pointing out the simplicity of the generative model of our method considering that those in prior work involve complicated graph search algorithm or alternating generation of atoms and bonds with multiple networks.

2.2 Learning Algorithm

Suppose we observe training examples $(x_i, i = 1, ..., n)$. The log-likelihood function is

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{4}$$

where $\theta = (\alpha, \beta)$. The learning gradient can be calculated according to

$$\nabla_\theta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z)\right] = \mathrm{E}_{p_\theta(z|x)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))\right]. \tag{5}$$

For the prior model, $\nabla_\alpha \log p_\alpha(z) = \nabla_\alpha f_\alpha(z) - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$. Thus the learning gradient for an example x is

$$\delta_\alpha(x) = \nabla_\alpha \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. \tag{6}$$

α is updated based on the difference between z inferred from empirical observation x, and z sampled from the current prior model.

For the generative model,

$$\delta_\beta(x) = \nabla_\beta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)], \tag{7}$$

where $\sum_{t=1}^{T} \log p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$ for text modeling which is about the reconstruction error.

Expectations in (6) and (7) require MCMC sampling of the prior model $p_\alpha(z)$ and the posterior distribution $p_\theta(z|x)$. Instead of learning a separate network for approximate inference, we follow [7] and use Langevin dynamics for short run MCMC which iterates:

$$z_0 \sim p_0(z),\ z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k,\ k = 1, ..., K. \tag{8}$$

where we initialize the dynamics from the fixed prior distribution of z, i.e., $p(z) \sim \mathrm{N}(0, I_d)$ and $\epsilon_k \sim \mathrm{N}(0, I_d)$ is the Gaussian white noise. $\pi(z)$ can be either $p_\alpha(z)$ or $p_\theta(z|x)$. In either case, $\nabla_z \log \pi(z)$ can be efficiently computed by back-propagation. The dynamics runs a fixed number of K steps with step size s.Denote the distribution of z_K to be $\tilde{\pi}(z)$. As shown in [1], the Kullback-Leibler divergence $D_{KL}(\tilde{\pi}\|\pi)$ decreases to zero monotonically as $K \to \infty$.

Specifically, denote the distribution of z_K to be $\tilde{p}_\alpha(z)$ if the target $\pi(z) = p_\alpha(z)$, and denote the distribution of z_K to be $\tilde{p}_\theta(z|x)$ if $\pi(z) = p_\theta(z|x)$. The learning gradients in equations (6) and (7) are modified to

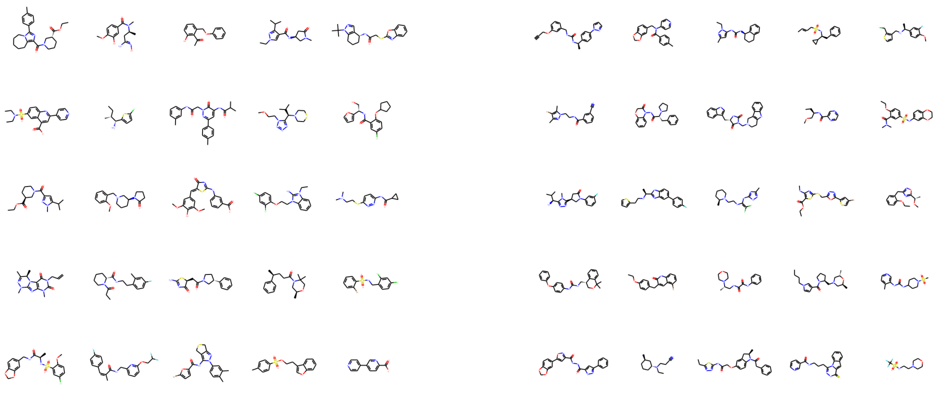
$$\tilde{\delta}_\alpha(x) = \mathrm{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)], \tag{9}$$

$$\tilde{\delta}_\beta(x) = \mathrm{E}_{\tilde{p}_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. \tag{10}$$

We then update α and β based on (9) and (10), where the expectations can be approximated by Monte Carlo samples. See [7] for theoretical foundation of the resulting learning algorithm. The short-run MCMC is efficient and mixes well in latent space due to the relative low-dimensionality of the latent space.

3 Experiments

A standard molecule dataset, ZINC [3], is used in our experiments. The latent space dimension is 32. The latent space energy-based model is implemented with a three-layer MLP with hidden dimension 200. The generator is a single layer LSTM with a hidden dimension of 1024 and the embedding dimension is 512. Figure 1 shows sample molecules generated from the data and randomly generated from our model.



(a) ZINC

(b) Generated

Figure 1: Sample molecules taken from the ZINC dataset (a) and generated by our model (b).

3.1 Validity, novelty, and uniqueness

We evaluate our model with three commonly used metrics: 1) validity, the percentage of valid molecules among all the generated ones; 2) novelty, the percentage of generated molecules not appearing in training set; 3) uniqueness, the percentage of unique ones among all the generated molecules. All metrics are computed based on 10,000 randomly generated molecules. Our model greatly improve previous LM-based models on validity and uniqueness and are competitive with fragment-based model and graph-based models using valency check. It is interesting to notice that the state-of-the-art graph-based models such as GCPN [13] and GraphAF [10], generate molecules with low validity rates if valency check is not applied. It appears that the graph-based models do not capture the chemical rules but instead strongly relies on explicit constraints. In contrast, our model is able to automatically learn the rules from the data.

Model	Model Family	Validity w/ check	Validity w/o check	Novelty	Uniqueness
GraphVAE (Simonovsky et al., 2018)	Graph	0.140	-	1.000	0.316
CGVAE (Liu et al., 2018)	Graph	1.000	-	1.000	0.998
GCPN (You et al., 2018)	Graph	1.000	0.200	1.000	1.000
NeVAE (Samanta et al., 2019)	Graph	1.000	-	0.999	1.000
MRNN (Popova et al., 2019)	Graph	1.000	0.650	1.000	0.999
GraphNVP (Madhawa et al., 2019)	Graph	0.426	-	1.000	0.948
GraphAF (Shi et al., 2020)	Graph	1.000	0.680	1.000	0.991
ChemVAE (Gomez-Bombarelli et al., 2018)	LM	0.170	-	0.980	0.310
GrammarVAE (Kusner et al., 2017)	LM	0.310	-	1.000	0.108
SDVAE (Dai et al., 2018)	LM	0.435	-	-	-
FragmentVAE (Podda et al., 2020)	LM	**1.000**	-	0.995	0.998
Ours	LM	0.955	-	**1.000**	**1.000**

Table 1: Performance obtained by our model against LM-based and graph-based baselines.

3.2 Molecular properties of samples

If a model distribution matches the data distribution well, marginal distributions of any statistics would also match. Three properties are critical for molecule modeling, especially in *de novo* drug design: 1) octanol/water partition coefficient (logP) which measures solubility; 2) quantitative estimate of drug-likeness (QED); 3) synthetic accessiblity score (SAS) which measures ease of synthesis. Each property can be viewed a statistic of the molecule data. In Figure 2, we compare the distributions of the three properties based on 10,000 samples from the data and our model. The distributions based on FragmentVAE are also included for a reference. It is clear that our model produces distributions close to data property distributions, even though there is not any explicit supervision given for learning the three molecular properties. Also, our model evidently improve over FragmentVAE in this regard.

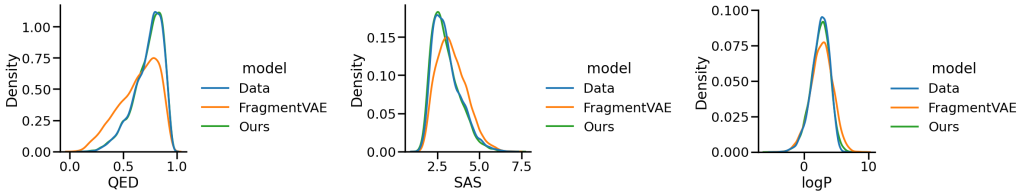


Figure 2: Distributions of molecular properties of data and 10,000 random samples from FragmentVAE and our model.

4 Conclusion

This work proposes to jointly learn a latent space energy-based prior model and a simple autoregressive generator for molecule modeling. Our approach yields a simple yet highly expressive model. The learned model generates valid and unique molecules with character-level SMILES representation. Key chemical properties of the generated samples closely resemble those of the data on a distribution level. These results provide strong evidence that the proposed model is able to automatically learn complicated chemical rules implicitly from the data.

References

[1] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[2] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[3] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[5] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954, 2017.

[6] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, pages 7795–7804, 2018.

[7] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *arXiv preprint arXiv:2006.08205*, 2020.

[8] Marco Podda, Davide Bacciu, and Alessio Micheli. A deep generative model for fragment-based molecule generation. *arXiv preprint arXiv:2002.12826*, 2020.

[9] Bidisha Samanta, DE Abir, Gourhari Jana, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez Rodriguez. Nevae: A deep generative model for molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1110–1117, 2019.

[10] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

[11] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer, 2018.

[12] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[13] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pages 6410–6421, 2018.