A Probabilistic Model for Molecular Geometry Generation and Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generating equilibrium 3D structures for molecular graphs remains a challenging task, which traditionally involves computationally expensive molecular dynamics simulations. We propose a probabilistic model for molecular conformation generation and geometry optimization based on graph neural networks and continuous normalizing flows. The model takes molecular graphs as input and learns the distributions *w.r.t.* inter-atomic distances. Based on the model, we formulate an algorithm for fast conformation sampling. In addition, the continuity of the learned distributions enables the model to perform gradient-based geometry optimization. Experiments show that the proposed model is competitive compared to recent state-of-the-arts.

1 Introduction

Recently, many deep learning methods for molecule generation have emerged. These methods operate using graph representations of molecules, where nodes represent atoms and edges represent bonds. However, a more intrinsic representation of a molecule is its 3D structure, commonly known as molecular geometry or conformation, which represents a molecule as a set of points. To bridge the gap between graph representations and 3D representations, we should consider generating valid and stable geometries of a given molecular graph, which remains a challenging task. Traditionally, molecular conformation generation is done by computationally expensive molecular dynamics [1, 2].

Machine learning methods have recently shown great potential for efficient molecular geometry generation and optimization by training on a collection of data to model the distribution of conformations based on a molecular graph. For example, [3] proposed using variational auto-encoders to generate 3D coordinates of atoms in a given molecular graph. The major limitation of this approach is that by directly generating the 3D coordinates of atoms it fails to model the rotational and translational invariance of molecular conformations. To address the issue of roto-translational invariance, instead of generating 3D coordinates, [4] proposed GraphDG, which employs VAEs to model the molecule's inter-atomic distances, and then generate the conformation using a distance geometry algorithm [5, 6]. Some other works [7–11] also focus on generating 3D structures but concentrate on the protein folding problem. They are not transferable to general molecules which have branched structures.

Inspired by recent progress in deep generative models, in this paper, we present a novel probabilistic model for molecular geometry generation and optimization. Specifically, we propose the *edge-conditioned continuous normalizing flow* (EC-CNF). The model first encodes edges in a molecular graph using graph neural networks [12, 13], then it employs normalizing flows to parameterize the distribution *w.r.t.* edge lengths conditioned on the edge features produced by the graph neural network. Based on the model, we formulate the algorithms for conformation sampling and geometry optimization. Experiments show that our model outperforms recent state-of-the-arts.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

2 Method

2.1 Overview

Notations A molecule is represented as an *extended* undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is the set of nodes representing atoms. $\mathcal{E}$ is the set of edges representing inter-atomic bonds and virtual edges including angular edges and dihedral edges [4]. Each node v in $\mathcal{V}$ is labeled with atomic properties such as element type. The edge in $\mathcal{E}$ connecting u and v is denoted as e_{uv}, and is labeled with its bond type. For the molecular 3D representation, each atom in $\mathcal{V}$ is assigned with a 3D position vector $\boldsymbol{r} \in \mathbb{R}^3$. We denote $d_{uv} = \|\boldsymbol{r}_u - \boldsymbol{r}_v\|_2$ as the Euclidean distance between the u^{th} and v^{th} atom. For brevity, we can represent all the positions $\{\boldsymbol{r}_v\}_{v\in\mathcal{V}}$ as a matrix $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{V}|\times 3}$ and all the distances between connected nodes $\{d_{uv}\}_{e_{uv}\in\mathcal{E}}$ as a vector $\boldsymbol{d} \in \mathbb{R}^{|\mathcal{E}|}$.

Problem Definition The goal of molecular geometry generation and optimization is modeling the distribution *w.r.t.* to 3D coordinates given a molecular graph $\mathcal{G}$, *i.e.*, $p(\boldsymbol{R}|\mathcal{G})$. However, to respect roto-translational invariance, we consider first modeling $p(\boldsymbol{d}|\mathcal{G})$, and then define $p(\boldsymbol{R}|\mathcal{G})$ based on $p(\boldsymbol{d}|\mathcal{G})$.

Overview of the Model We model $p(\boldsymbol{d}|\mathcal{G})$ using two components: First, we employ graph neural networks [12, 14] to encode the structural context of each edge in $\mathcal{G}$. Then, we use conditional continuous normalizing flows [15–18] to parameterize the distribution *w.r.t.* to the length of each edge conditioned on the encoded context. To generate molecular geometry, we define $p(\boldsymbol{R}|\mathcal{G})$ in an auto-regressive fashion based on $p(\boldsymbol{d}|\mathcal{G})$. To perform geometry optimization, we use $p(\boldsymbol{d}|\mathcal{G})$ to define an energy function.

2.2 Model

Structural Encoding with MPNNs The distribution of distances $p(\boldsymbol{d}|\mathcal{G})$ is conditioned on the structural context of each edge in the graph $\mathcal{G}$. Therefore, to encode the structural context, we use RGCNs [13] that take the graph $\mathcal{G}$ as input and output *nodewise* features $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}|\times F}$:

$$\boldsymbol{H} = \mathrm{RGCN}(\mathcal{G}). \tag{1}$$

To obtain *edge features*, we apply the following transformation:

$$\boldsymbol{y}_{uv} = \mathrm{MLP}([\boldsymbol{H}_u \odot \boldsymbol{H}_v \| \boldsymbol{\beta}_{uv}]), \tag{2}$$

where $\boldsymbol{y}_{uv}$ is the feature of edge (u, v), $\boldsymbol{H}_u$ and $\boldsymbol{H}_v$ are node features of node u and v respectively, and $\boldsymbol{\beta}_{uv}$ is the embedding of edge (u, v) which contains information such as bond types.

Distribution Fitting with the EC-CNF We propose the *edge-conditioned continuous normalizing flow* (EC-CNF) to model the distributions *w.r.t.* edge lengths $p(\boldsymbol{d}|\mathcal{G})$. Specifically, we use the CNF to model the distribution *w.r.t.* distance of each edge:

$$d_{uv} = F(z(t_0), \boldsymbol{y}_{uv}) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \boldsymbol{y}_{uv})\mathrm{d}t, \tag{3}$$

where $z(t_0) \sim \mathcal{N}(0, 1)$ is the source distribution, f is the dynamic that transforms $z(t_0)$ to the target distribution. We point out that, despite the distributions are separately modeled, they are aware of the molecular graph's topological structure and they vary depending on the structural environment of edges. Thus they are inter-related through the graph $\mathcal{G}$. Based on the CNF, the target distribution *w.r.t.* d_{uv} conditioned on the edge features $\boldsymbol{y}_{uv}$ is:

$$\log p(d_{uv}|\boldsymbol{y}_{uv}) = \log p(F^{-1}(d_{uv}, \boldsymbol{y}_{uv})) - \int_{t_0}^{t_1} \mathrm{Tr}\Big(\frac{\partial f}{\partial z(t)}\Big)\mathrm{d}t. \tag{4}$$

The distribution is easy to sample from and the probability is computable with a black-box ODE solver [15, 16].

Training Objective Training the model amounts to maximizing the log-likelihood function:

$$\mathcal{L} = \mathbb{E}_{(\mathcal{G},\boldsymbol{R})\sim p_{\text{data}}}\Big[\sum_{(u,v)\in\mathcal{E}} \log p\big(\|\boldsymbol{R}_u - \boldsymbol{R}_v\|_2 \big| \boldsymbol{y}_{uv}(\mathcal{G})\big)\Big], \tag{5}$$

where $(\mathcal{G}, \boldsymbol{R})$ is a molecular graph-geometry pair drawn from the data distribution p_{data}.

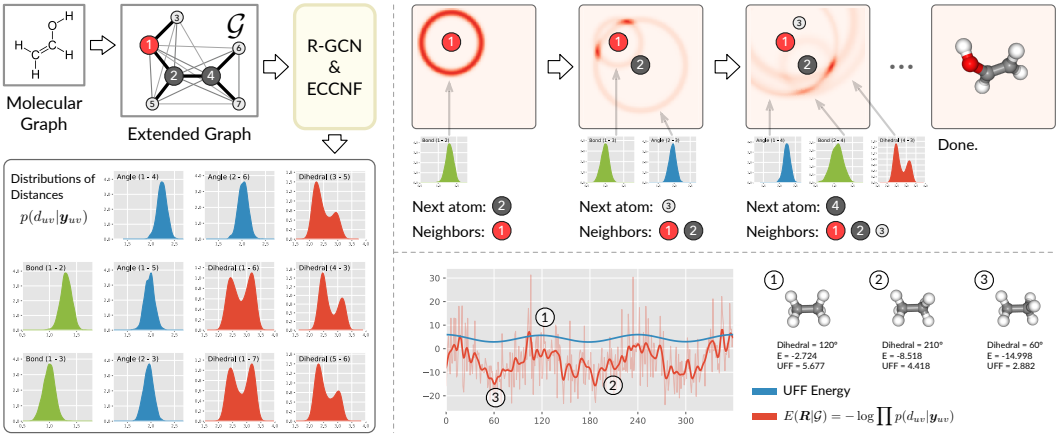

Figure 1: **(a) Left**: An illustration of our model. The model takes extended molecular graphs as input and leads to a set of distributions *w.r.t.* to distances. **(b) Top right**: An illustration of the auto-regressive conformation generation process. Atoms are placed one-by-one according to the probability distribution $p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1}, \mathcal{G})$ which is based on $p(d_{uv}|\boldsymbol{y}_{uv})$. **(c) Bottom right**: Energy function defined in Eq. 7. Our energy function is positively correlated to the real molecular potential energy function, where equilibrium conformations are at lower energy levels.

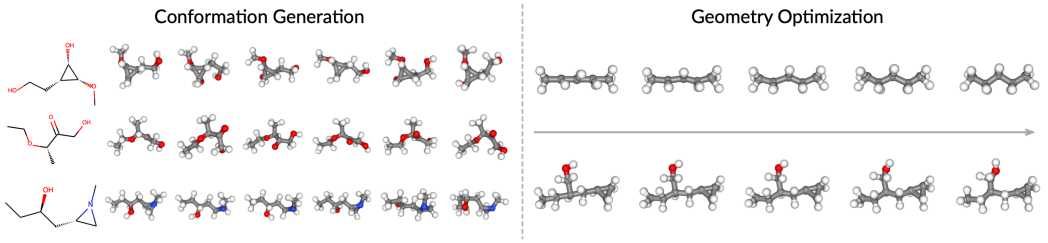

Figure 2: **(a) Left**: Generated conformations from our model. **(b) Right**: Visualization of the geometry optimization process.

2.3 Algorithms

Conformation Generation Assume the nodes in the graph $\mathcal{G}$ are sorted according to the BFS order, denoted as $u_1, \ldots, u_N$, where $N = |\mathcal{V}|$. Generating 3D coordinates is thus done by sampling progressively from the following auto-regressive distribution:

$$p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1}, \mathcal{G}) = \prod_{\substack{j<i \\ (u_i,u_j)\in\mathcal{E}}} p\left(\|\boldsymbol{R}_i - \boldsymbol{R}_j\|_2 \middle| \boldsymbol{y}_{u_i u_j}\right), \quad \boldsymbol{R}_1 = \boldsymbol{0}, \tag{6}$$

where $\boldsymbol{R}_i$ is the 3D coordinate of the node u_i. With the auto-regressive distribution formulated, the distribution *w.r.t.* conformations is naturally defined as $p(\boldsymbol{R}|\mathcal{G}) = \prod p(\boldsymbol{R}_i|\boldsymbol{R}_{1:i-1}, \mathcal{G})$.

Although the auto-regressive approach is similar to G-SchNet [19], our model is different from G-SchNet in at least two aspects: first, we focus on conformation generation and geometry optimization, while G-SchNet focuses on unconditional generation; second, we employ conditional normalizing flows to parameterize the distribution of distances, making the probability continuous, which is the foundation for geometry optimization, while G-SchNet discretizes such distributions.

Geometry Optimization We define an energy function based on the distributions *w.r.t.* distances:

$$E(\boldsymbol{R}|\mathcal{G}) = -\log \prod_{(u,v)\in\mathcal{E}} p\left(\|\boldsymbol{R}_u - \boldsymbol{R}_v\|_2 \middle| \boldsymbol{y}_{uv}\right). \tag{7}$$

By performing gradient descent on the energy function, we can optimize the molecular geometry.

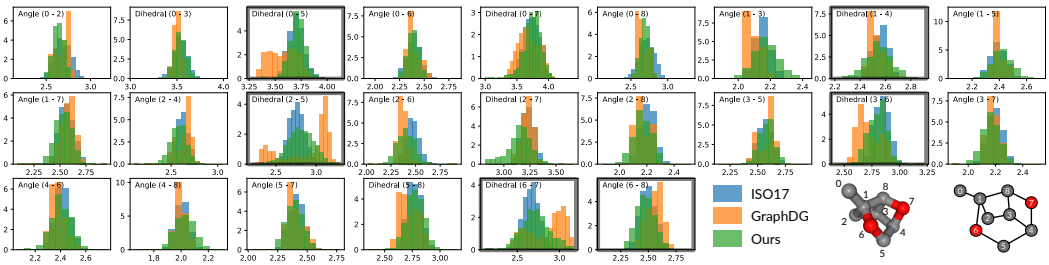

Figure 3: Marginal distributions $p(d_{uv}|\mathcal{G})$ of ground-truth (ISO17) and generated conformations from GraphDG and our model. We omit the distances of bonded atom-pairs which are trivial to model, and concentrate on angular and dihedral atom-pairs. Our model fits the target distribution better than GraphDG does, especially on the atom-pairs highlighted by black frames. (Note that following [4], the distances are computed from the generated 3D structures, not directly sampled from EC-CNF.)

Table 1: Assessment of the accuracy of the distributions over distances generated compared to the ground-truth. We evaluate the distance distribution of single ($p(d_{uv}|\mathcal{G})$), pair ($p(d_{uv}, d_{ij}|\mathcal{G})$) and all ($p(\boldsymbol{d}|\mathcal{G})$) edges between heavy atoms. Molecular graphs $\mathcal{G}$ are taken from the test set of ISO17. **Median** and **mean** MMDs between the ground truth and generated distributions are reported.

	Single		Pair		All	
	Mean	Median	Mean	Median	Mean	Median
RDKit [20]	3.4513	3.1602	3.8452	3.6287	4.0866	3.7519
CVGAE [3]	4.1789	4.1762	4.9184	5.1856	5.9747	5.9928
GraphDG [4]	0.7645	0.2346	0.8920	0.3287	1.1949	0.5485
Ours	**0.5042**	**0.1627**	**0.6086**	**0.2150**	**1.0239**	**0.5172**

3 Experiments

3.1 Conformation Generation

We follow the setting (datasets and metrics) of GraphDG [4] to evaluate our model, and compute the MMDs (maximum mean discrepancy) of the distribution of all the distances between heavy atoms $p(\boldsymbol{d}|\mathcal{G})$, the MMDs of pair distances $p(d_{uv}, d_{ij}|\mathcal{G})$ and the MMDs of single distances $p(d_{uv}|\mathcal{G})$. As shown in Table 1, our method outperforms other methods. We also plot the distributions in Figure 3 for comparison. Additionally, we train our model using 50000 molecule-conformation pairs in the QM9 dataset [21] and generate conformations for various molecules in the rest of the QM9 dataset with the model. We present some generated conformations in Figure 2. The visual result shows that our model is capable to discover diverse conformations.

3.2 Geometry Optimization

We use the model trained on the QM9 dataset to perform geometry optimization and present the trajectories of two molecules in Figure 2. The trajectories show that our model is able to optimize molecular geometry towards equilibrium states. We also plot the energy function (defined in Eq. 7) of CH_6 in Figure 1. The plot indicates that the energy function is positively correlated with the real potential energy function (approximated by the UFF [22, 23]), where stable rotamers are at lower energy levels.

4 Conclusions and Future Work

In this paper, we propose a probabilistic model for molecular conformation generation and geometry optimization. Experiments show that our model outperforms the state-of-the-art model GraphDG. Future work includes adapting the model for larger molecular structures such as drugs and proteins.

References

[1] Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9): 4035–4061, 2016.

[2] Andrew J Ballard, Stefano Martiniani, Jacob D Stevenson, Sandeep Somani, and David J Wales. Exploiting the potential energy landscape to sample free energy. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(3):273–289, 2015.

[3] Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *arXiv preprint arXiv:1904.00314*, 2019.

[4] Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. In *International Conference on Machine Learning*, 2020.

[5] Leo Liberti, Carlile Lavor, Nelson Maculan, and Antonio Mucherino. Euclidean distance geometry and applications. *SIAM review*, 56(1):3–69, 2014.

[6] Gordon M Crippen, Timothy F Havel, et al. *Distance geometry and molecular conformation*, volume 74. Research Studies Press Taunton, 1988.

[7] Tobias Lemke and Christine Peter. Encodermap: Dimensionality reduction and generation of molecule conformations. *Journal of chemical theory and computation*, 15(2):1209–1215, 2019.

[8] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8 (4):292–301, 2019.

[9] John Ingraham, Adam J Riesselman, Chris Sander, and Debora S Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations*, 2019.

[10] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[11] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[13] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[16] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[17] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[18] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[19] Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In *Advances in Neural Information Processing Systems*, pages 7566–7578, 2019.

[20] Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55 (12):2562–2574, 2015.

[21] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[22] Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

[23] Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.