Updates and Progress

Yogesh Verma
Doctoral Candidate
Aalto University

Updates

- Paper read: 23/1280
  - Deconstructing Inductive bias of HNN
  - PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows
  - Continuous Graph Flow
  - GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation [Reading]
  - Spanning Tree-based Graph Generation for Molecules [Reading]

Updates

- Paper read: 23/1280
  - Deconstructing Inductive bias of HNN
  - PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows
  - Continuous Graph Flow
  - GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation [Reading]
  - Spanning Tree-based Graph Generation for Molecules [Reading]
- CNF for generating valid molecules
- Molecular surface by 3D Zerneike Descriptors

ICLR'22 Summary

- Data-Efficient Graph Grammar Learning for Molecular Generation
- GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation
- Amortized Tree Generation for Bottom-up Synthesis Planning and Synthesizable Molecular Design
- Iterative Refinement Graph Neural Network for Antibody Sequence-Structure Co-design
- Independent SE(3)-Equivariant Models for End-to-End Rigid Protein Docking
- Spanning Tree-based Graph Generation for Molecules
- GeneDisco: A Benchmark for Experimental Design in Drug Discovery
- Spatial Graph Attention and Curiosity-driven Policy for Antiviral Drug Discovery
- Learning to Extend Molecular Scaffolds with Structural Motifs
- Pre-training Molecular Graph Representation with 3D Geometry
- Differentiable Scaffolding Tree for Molecule Optimization
- Geometric Transformers for Protein Interface Contact Prediction
- Learning 3D Representations of Molecular Chirality with Invariance to Bond Rotations
- Top-N: Equivariant Set and Graph Generation without Exchangeability
- Chemical-Reaction-Aware Molecule Representation Learning
- Energy-Inspired Molecular Conformation Optimization
- An Autoregressive Flow Model for 3D Molecular Geometry Generation from Scratch
- Spherical Message Passing for 3D Molecular Graphs
- Maximum n-times Coverage for Vaccine Design

Molecule as a 2-D Graph

- Given a molecule with a graph representation (connectivity C) $\mathcal{G} = (V, E)$, one can define a probability distribution at each node over the atomic vocabulary (Vocabulary of all the unique atoms spanning the whole molecule data-set) conditional over each node(atom) neighbours.

Background

Molecule as a 2-D Graph

- Given a molecule with a graph representation (connectivity C) $\mathcal{G} = (V, E)$, one can define a probability distribution at each node over the atomic vocabulary (Vocabulary of all the unique atoms spanning the whole molecule data-set) conditional over each node(atom) neighbours.

The total distribution for the whole graph can be written as

$$P(X) = \Pi_i^{N(V)} p(x_i | N_{J(i)}) \tag{1}$$

- This appears to be more intuitive as in molecules each atom in a bond must follow certain conditions imposed by neighbouring atoms like valency, bond type (atoms which can pair with double bond, single bond etc.), etc; thus imposing some constraints over the atom assignment which may lead to generating valid molecules.

Background: Continuous Normalizing flows

Continuous normalizing flows (CNFs) model the continuous-time dynamics. Given a random variable z, the following defines the change in the state of the variable

Background: Continuous Normalizing flows

Continuous normalizing flows (CNFs) model the continuous-time dynamics. Given a random variable z, the following defines the change in the state of the variable

$$\frac{\partial z}{\partial t} = f(z(t), t) \tag{2}$$

The dynamics of the log-probability of a random variable is then defined as the following ODE

$$\frac{\partial \log p(z(t))}{\partial t} = -Tr(\frac{\partial f}{\partial z(t)}) \tag{3}$$

Following the above equation, the log-probability of the variable z at time t_1 starting from time t_0 is

$$log(p(z(t_1))) = log(p(z(t_0))) - \int_{t_0}^{t_1} Tr(\frac{\partial f(z(t), \theta)}{\partial z(t)})dt \tag{4}$$

Flowing Molecular Graphs

- Building on CNFs, we present flows on graphs specially applied in the molecular regime where we model the continuous time dynamics of random variables on graphs with respect to some conditionals over connectivity of the graph applied to graph structured data

Flowing Molecular Graphs

- Given a set of vertices $\mathbf{X}$ for a graph (molecule), the goal is to learn the joint distribution $P(X)$ given by Eq.1 of the set of nodes.
- For continuous time dynamics of the set of variables X, by following Eq. 3,4 we formulate an ODE system as follows

$$\begin{bmatrix} \dot{\mathbf{log}}(\mathbf{p}(\mathbf{x_1(t)}|\mathbf{N_{J(1)}})) \\ \dot{\mathbf{log}}(\mathbf{p}(\mathbf{x_2(t)}|\mathbf{N_{J(2)}})) \\ \vdots \\ \dot{\mathbf{log}}(\mathbf{p}(\mathbf{x_n(t)}|\mathbf{N_{J(n)}})) \end{bmatrix} = \begin{bmatrix} -Tr(\frac{\partial \mathbf{f(X(t)}, \mathbf{N_{J(1)}}}{\partial \mathbf{x_1(t)}}) \\ -Tr(\frac{\partial \mathbf{f(X(t)}, \mathbf{N_{J(2)}}}{\partial \mathbf{x_2(t)}}) \\ \vdots \\ -Tr(\frac{\partial \mathbf{f(X(t)}, \mathbf{N_{J(n)}}}{\partial \mathbf{x_n(t)}}) \end{bmatrix} \tag{5}$$

where each $\mathbf{x_i(t)}$ is i^{th} node, $\mathbf{N_{J(i)}}$ its neighbours and $\mathbf{x_i} \in \mathbf{X}$.

Flowing Molecular Graph

Following the Eq.6, the change in log probability can also be represented as

$$log(p(\mathbf{x_i}(\mathbf{t})|N_{J(i)})) = log(p(\mathbf{x_i}(\mathbf{0})|N_{J(i)})) - \int_0^t Tr(\frac{\partial f(\mathbf{X}(\mathbf{t}), N_{J(i)}, \theta)}{\partial \mathbf{x_i}(\mathbf{t})})dt$$

Flowing Molecular Graph

Following the Eq.6, the change in log probability can also be represented as

$$log(p(\mathbf{x_i}(\mathbf{t})|N_{J(i)})) = log(p(\mathbf{x_i}(\mathbf{0})|N_{J(i)})) - \int_0^t Tr(\frac{\partial f(\mathbf{X}(\mathbf{t}), N_{J(i)}, \theta)}{\partial \mathbf{x_i}(\mathbf{t})})dt$$

- The above equation forms a set of coupled equations as all the corresponding node densities are coupled with their neighbours and flowing together. This is the conditional extension of CNFs as we are dealing with conditional distributions which are inducing coupling between the flows.
- The above coupled equations provide constraints over the choice of function f

Defining f

- The above flowing mechanism is defined with the help of function $f(z_i, N_{J(i)}, \theta)$, to which the input are the nodes and its neighbours and is described by parameters θ. We define a specific choice of f as

$$f(x_i, N_{J(i)}, \theta = \{\theta_{ij}\}) = \sum_{j}^{N_{J(i)}} \theta_{ij}\phi(x_i, x_j) \tag{6}$$

- where $\phi(x_i, x_j)$ are the radial basis functions and θ_{ij} are the parameters. The task now reduces to learn the distribution of parameters θ_{ij}, given the target distribution on nodes of molecules.

Optimization

- Minimize the $D_{KL}\ [p(x(t)|N_J, \theta)\ ||\ p(x^*|N_J)]$ to fit the flow based model, which can be formally written as

$$\mathcal{L}(\theta) = D_{KL}\ [p(x(t)|N_J, \theta)\ ||\ p(x^*|N_J)] \tag{7}$$
$$= \mathbb{E}_{p(x(t)|N_J,\theta)}\ [\log(p(x(t)|N_J, \theta)) - \log(p(x^*|N_J)] \tag{8}$$

- θ are parameters of function and $p(\mathbf{x}^*|N_J)$ is the target distribution.

Optimization

- Minimize the $D_{KL}\ [p(x(t)|N_J, \theta)\ ||\ p(x^*|N_J)]$ to fit the flow based model, which can be formally written as

$$\mathcal{L}(\theta) = D_{KL}\ [p(x(t)|N_J, \theta)\ ||\ p(x^*|N_J)] \tag{7}$$
$$= \mathbb{E}_{p(x(t)|N_J,\theta)}\ [\log(p(x(t)|N_J, \theta)) - \log(p(x^*|N_J)] \tag{8}$$

- θ are parameters of function and $p(\mathbf{x}^*|N_J)$ is the target distribution. The $p(x(t)|N_J, \theta)$ can be represented in terms of base distribution as

$$log(p(\mathbf{x_i(t)}|N_{J(i)})) = log(p(\mathbf{x_i(0)}|N_{J(i)})) - \int_0^t Tr(\frac{\partial f(\mathbf{X(t)}, N_{J(i)}, \theta)}{\partial \mathbf{x_i(t)}})dt$$

- So, the Eq. 7 transforms the expectation over the base distribution from where we could sample from the base density and flow it using CNF to get the final density.

Parametrization for initial density

- One can extend the optimization criteria when the base distribution ($p(\mathbf{x_i}(\mathbf{0})|N_{J(i)})$) is parametrized by parameters ψ and a similar approach like variational auto-encoder formalism can be followed

$$\mathcal{L}(\theta, \phi) = -\mathrm{E}_{x(0)\sim q_\psi(x(0)|x^*)}[\log p_\theta(x(t)|x(0)] + \mathrm{KL}[q_\psi(x(0)|x^*)||p(z)] \tag{9}$$

- The $p_\theta(x(t)|x(0)]$ is represented by CNF as described in previous sections and $q_\psi(x(0)|x^*)$ is the base distribution conditioned on data samples (encoder distribution as in case of VAE)

3D Zernieke Descriptors/Moments (3DZD)

- The 3DZD is a mathematical series expansion of 3D function, which project a 3D object to a compact representation.
- These moments are computed as a projection of the function defining the object onto a set of orthonormal functions within the unit ball – the 3D Zernike polynomials

3D Zernieke Descriptors/Moments (3DZD)

- The 3DZD is a mathematical series expansion of 3D function, which project a 3D object to a compact representation.
- These moments are computed as a projection of the function defining the object onto a set of orthonormal functions within the unit ball – the 3D Zernike polynomials
- The 3D Zernike polynomials defined on order n, degree l, and repetition m, are given by

$$Z_{nl}^{m}(r, v, \phi) = R_{nl}(r) Y_{l}^{m}(v, \phi)$$

where

$$-l < m < l,\ 0 \leq l \leq n$$

$$Y_{l}^{m}(v, \phi) \ \rightarrow \text{Spherical harmonics}$$

$$R_{nl}(r) \ = r^{l} \sum_{v=0}^{k} q_{kl}^{v} r^{2v} (\text{Radial polynomials})$$

3D Zernieke Descriptors (3DZD)

$$q_{kl}^{v} = \frac{(-1)^k}{2^{2k}} \sqrt{\frac{2l+4k+3}{3}} \binom{2k}{k} (-1)^v \cdot \frac{\binom{k}{v} \binom{2(k+l+v)+1}{2k}}{\binom{k+l+v}{k}}$$

These are determined to guarantee the orthonormality of the functions in the unit ball.

Representation

- Let $f(\mathbf{x})$ is a 3D function, representing the molecule/protein (e.g: $\mathbf{x} = (x,y,z)$ $f(\mathbf{x})$, should be the description of the surface shape, which can be computed by placing the protein/molecule structure onto a 3D grid)

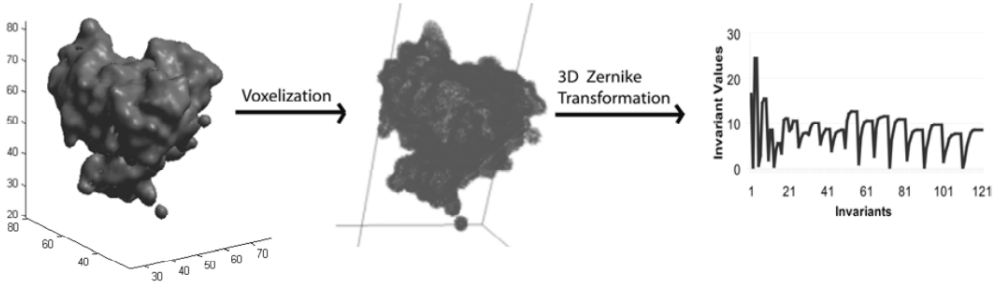
Representation

- Let $f(\mathbf{x})$ is a 3D function, representing the molecule/protein (e.g: $\mathbf{x}$ = (x,y,z) $f(\mathbf{x})$, should be the description of the surface shape, which can be computed by placing the protein/molecule structure onto a 3D grid)
- The 3D Zernike moments of $f(\mathbf{x})$ (conversion to Cartesian coordinates $Z_{nl}^{m}(\mathbf{x})$ using harmonic polynomials) are defined as the coefficients of the expansion in this orthonormal basis by

$$\Omega_{nl}^{m} = \frac{3}{4\pi} \int_{|\mathbf{x}|\leq 1} f(\mathbf{x}) \bar{Z}_{nl}^{m}(\mathbf{x}) d\mathbf{x} \tag{10}$$

$$\frac{3}{4\pi} \int_{\|\mathbf{x}\|\leq 1} Z_{nl}^{m}(\mathbf{x}) \cdot \overline{Z_{n'l'}^{m'}(\mathbf{x})} d\mathbf{x} = \delta_{nn'} \delta_{ll'} \delta^{mm'} \tag{11}$$

Representation



Reconstruction

Since the functions Z_{nl}^m form a complete orthonormal system, it is possible to approximate the original function $f(\mathbf{x})$ by a finite number of 3D Zernike moments (Ω_{nl}^m):

$$\hat{f}(\mathbf{x}) = \sum_n \sum_l \sum_m \Omega_{nl}^m \cdot Z_{nl}^m(\mathbf{x}) \tag{12}$$

Connection to 3D Molecule shape generation

- One has a flexibility to represent 3-D molecule in Cartesian or Spherical coordinates or a Mesh-representation

Connection to 3D Molecule shape generation

- One has a flexibility to represent 3-D molecule in Cartesian or Spherical coordinates or a Mesh-representation
- Starting from an input point cloud (X), an encoder $Q_\phi(z|X)$ infers a posterior over shape (moment) representations, samples a shape (moments) representation $z \sim Q_\phi(z|X)$, compute the reconstruction likelihood of X (with repect to its 3DZD/moments) (L_{reco}) through CNF G_θ conditioned on z. Model can be trained end-to-end to maximize the evidence lower bound (ELBO)

Connection to 3D Molecule shape generation

- One has a flexibility to represent 3-D molecule in Cartesian or Spherical coordinates or a Mesh-representation
- Starting from an input point cloud (X), an encoder $Q_\phi(z|X)$ infers a posterior over shape (moment) representations, samples a shape (moments) representation $z \sim Q_\phi(z|X)$, compute the reconstruction likelihood of X (with repect to its 3DZD/moments) (L_{reco}) through CNF G_θ conditioned on z. Model can be trained end-to-end to maximize the evidence lower bound (ELBO)
- During testing phase, we sample a moment/shape representation $\tilde{z}$ by sampling from $Q_\phi(z|X)$. To sample moments from the moment/shape represented by $\tilde{z}$, we first sample points from the 3-D Gaussian prior and then move them according to the CNF parameterized by $\tilde{z}$ to get final moments and reconstruct the surface using Eq.12

Connection to 3D Molecule shape generation

- One has a flexibility to represent 3-D molecule in Cartesian or Spherical coordinates or a Mesh-representation
- Starting from an input point cloud (X), an encoder $Q_\phi(z|X)$ infers a posterior over shape (moment) representations, samples a shape (moments) representation $z \sim Q_\phi(z|X)$, compute the reconstruction likelihood of X (with repect to its 3DZD/moments) (L_{reco}) through CNF G_θ conditioned on z. Model can be trained end-to-end to maximize the evidence lower bound (ELBO)
- During testing phase, we sample a moment/shape representation $\tilde{z}$ by sampling from $Q_\phi(z|X)$. To sample moments from the moment/shape represented by $\tilde{z}$, we first sample points from the 3-D Gaussian prior and then move them according to the CNF parameterized by $\tilde{z}$ to get final moments and reconstruct the surface using Eq.12
- It can also be extended to multi-dim like representing the activity, hydro-phobicity etc as 3-D surface and identifying the regions within generated molecules (or generating new molecules with prior over possible regions by using this)

Summer School

- ProbAI: Nordic Probabilistic AI School [13-17 June 2022] (by FCAI, Norwegian Open AI lab, etc.)
- LOGML (London Geometry and Machine Learning) Summer School 2022 [11-15 July 2022]

ProbAI

Topics covered are

- Probabilistic models, Deep generative models
- Variational approximations, latent variable models
- Normalizing flows, neural ODEs, probabilistic programming, and much more.

Link: `https://probabilistic.ai/`

LOGML

The London Geometry and Machine Learning Summer School 2022 (LOGML) aims to bring together geometers and machine learners to work together on a variety of problems. During the summer school, the attendees will each engage in a week-long research project working at the interface of machine learning and geometry. Topics covered are

- Graph Neural Networks (e.g. in bioinformatics, recommender systems. . .)
- Representation learning and geometry (e.g. hyperbolic embeddings, hyperbolic neural networks, mixed-curvature representations)
- Invariant and equivariant representations
- Optimal Transport and its applications and much more!

Link: `https://www.logml.ai/home`

THANK YOU
FEEDBACK?