Updates and Progress

Yogesh Verma
Doctoral Candidate
Aalto University

Updates

- Paper read: 16/1280
  - Your classifier is secretly an energy based model and you should treat it like one
  - Neural Relational Inference for Interacting systems
  - Learning neural generative dynamics for molecular conformation generation
  - Geometric Deep Learning on Molecular Representations
  - EQUIBIND: Geometric Deep Learning for Drug Binding Structure Prediction [Reading]

Updates

- Paper read: 16/1280
  - Your classifier is secretly an energy based model and you should treat it like one
  - Neural Relational Inference for Interacting systems
  - Learning neural generative dynamics for molecular conformation generation
  - Geometric Deep Learning on Molecular Representations
  - EQUIBIND: Geometric Deep Learning for Drug Binding Structure Prediction [Reading]
- Spectral Density over Graphs
- Conformer generation by Z-matrix

Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

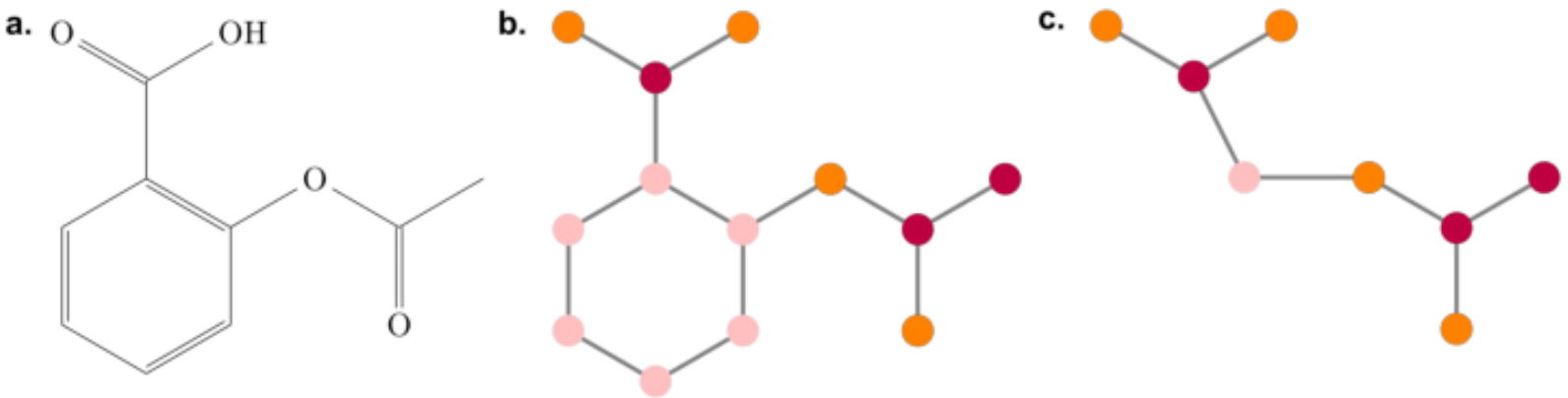
- Representation of subgraphs within molecule ?

Junction Tree

- A tree decomposition maps a graph $\mathcal{G}$ into a junction tree by contracting certain vertices into a single node.

Junction Tree

- A tree decomposition maps a graph $\mathcal{G}$ into a junction tree by contracting certain vertices into a single node.
- For a given graph $\mathcal{G}$, a junction tree $\mathcal{T}_{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ is a connected tree where $\mathcal{V} = (C_1, C_2, ..., C_n)$ and $\mathcal{E}$ are corresponding node and edge set.



Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

- Representation of subgraphs within molecule ?
- How to define spectral density?

Defining spectral density over junction tree

Can be done in many ways, but we propose following:

Defining spectral density over junction tree

Can be done in many ways, but we propose following:

- Inverse Mixture flows

Defining spectral density over junction tree

Can be done in many ways, but we propose following:

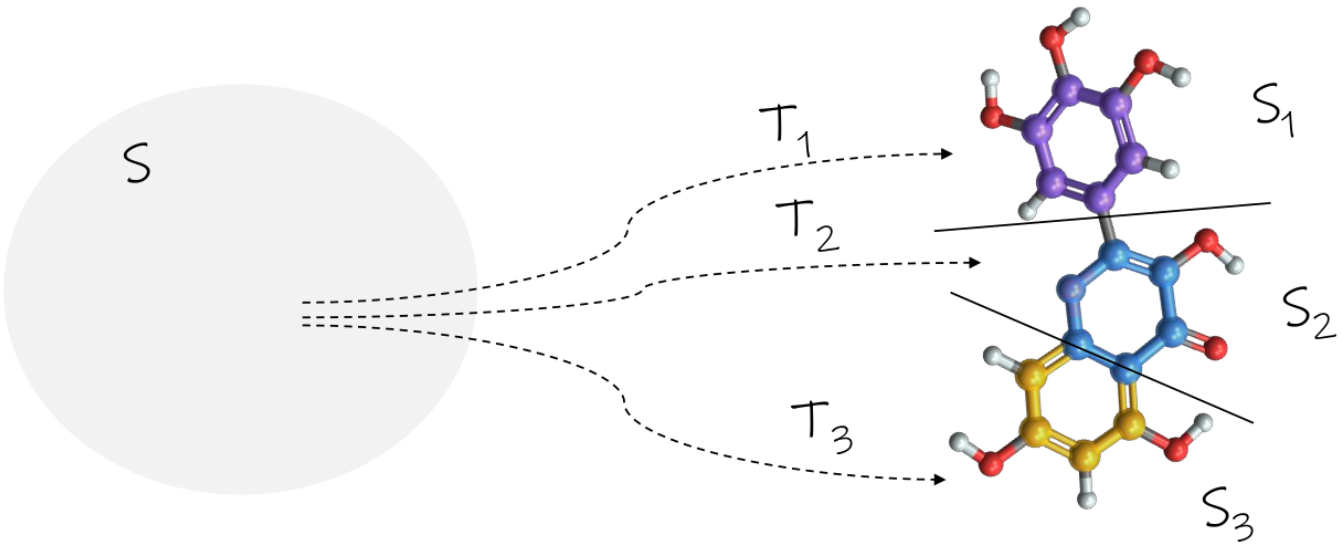
- Inverse Mixture flows
- Continuous Normalizing flows

Inverse Mixture flows

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph and its junction tree ($\mathcal{T}_{\mathcal{G}}$).

Inverse Mixture flows

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph and its junction tree ($\mathcal{T}_{\mathcal{G}}$).
- We define a base density ($\mathcal{S}$) from which spectral density over each node of junction tree ($\mathcal{S}_i$) can be computed via inverse mixture flow.



Inverse Mixture flows

- We can define a transformation $T_i : \mathcal{S} \to \mathcal{V}_i$, where T simply maps $\mathcal{S}$ to $T_i(\mathcal{S})_{i \in \mathcal{V}}$ giving a density over each node of $\mathcal{T}_\mathcal{G}$

Inverse Mixture flows

- We can define a transformation $T_i : \mathcal{S} \rightarrow \mathcal{V}_i$, where T simply maps $\mathcal{S}$ to $T_i(\mathcal{S})_{i \in \mathcal{V}}$ giving a density over each node of $\mathcal{T}_{\mathcal{G}}$
- Select $T_i(\mathcal{S})_{i \in \mathcal{V}}$ at random by $p(i|s)$, defining the input density as $p(s, i) = p_s(s)p(i|s)$. Finally, we get

$$p_v(v) = p_u(R(v))p(i(v)|R(v))|det\mathcal{J}_R(v)| \tag{1}$$

- where $i(v)$ indexes the subset of $\mathcal{V}$ in which v belongs and $T_i^{-1}(v) = R^{-1}(v)$.

Inverse Mixture flows

- We can define a transformation $T_i : \mathcal{S} \rightarrow \mathcal{V}_i$, where T simply maps $\mathcal{S}$ to $T_i(\mathcal{S})_{i \in \mathcal{V}}$ giving a density over each node of $\mathcal{T}_\mathcal{G}$
- Select $T_i(\mathcal{S})_{i \in \mathcal{V}}$ at random by $p(i|s)$, defining the input density as $p(s, i) = p_s(s)p(i|s)$. Finally, we get

$$p_v(v) = p_u(R(v))p(i(v)|R(v))|det\mathcal{J}_R(v)| \tag{1}$$

- where $i(v)$ indexes the subset of $\mathcal{V}$ in which v belongs and $T_i^{-1}(v) = R^{-1}(v)$.
- It is a mixture of flows, where i^{th} component uses transformation $T_i : \mathcal{S} \rightarrow \mathcal{V}_i$, base distribution $p_s(s|i) \propto p_s(s)p(i|s)$ and mixture weight $p(i) = \int p_s(s)p(i|s)ds$.

Continuous Normalizing flows

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph and its junction tree $(\mathcal{T}_{\mathcal{G}})$

Continuous Normalizing flows

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph and its junction tree $(\mathcal{T}_{\mathcal{G}})$
- Initialize a Gaussian density (multi-modal or uni-modal) over each node $\mathcal{V}_i$ of the junction tree $(\mathcal{T}_{\mathcal{G}})$

Continuous Normalizing flows

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph and its junction tree $(\mathcal{T}_{\mathcal{G}})$
- Initialize a Gaussian density (multi-modal or uni-modal) over each node $\mathcal{V}_i$ of the junction tree $(\mathcal{T}_{\mathcal{G}})$
- The transformation of the defined density will be governed by the CNF, where each node density is independent of other nodes density. The exact density can be calculated as:

$$log(\mathcal{S}_i) = log(z(t_0)) - \int_{t_0}^{t_1} \frac{\partial f_\theta}{\partial z(t)} \tag{2}$$

where $z(t_0)$ is an initial distribution and f_θ can be a transforming function parametrized by a BNN. The dynamics of the $z(t)$ are given by a continuous-time dynamics as:

$$\frac{\partial z(t)}{\partial t} = f_\theta(z(t), t) \tag{3}$$

Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

- Representation of subgraphs within molecule ?
- How to define spectral density?
- How to perform Fourier transform?

Graph Fourier Transform (1)

We now have the spectral density at each node of $\mathcal{T}_{\mathcal{G}}$, we sample $N_{\mathcal{V}}$ points from the distribution where $N_{\mathcal{V}}$ represents total number of nodes in $\mathcal{T}_{\mathcal{G}}$. We then perform an Inverse Graph Fourier Transform (GFT) given as

$$f_i(k) = \sum_{l=0}^{N_{\mathcal{V}}-1} \mathcal{S}_i(l) u_l(k) \tag{4}$$

where $\mathcal{S}_i$ is the density for i^{th} node, k is the index of the node for which we are computing the value and u_l are the eigen vectors of Graph Laplacian ($\mathbf{L}$) defined on $\mathcal{T}_{\mathcal{G}}$. If the graph has N nodes, then the $\mathbf{L}$ will be a $N \times N$ matrix and has N eigenvectors corresponding to each node.

Graph Fourier Transform (2)

$$f_i(k) = \sum_{l=0}^{N_\mathcal{V}-1} \mathcal{S}_i(l) u_l(k) \tag{5}$$

- The eq. 6 returns a distinct value at k^{th} node of $\mathcal{T}_\mathcal{G}$ $(f_i(k))$ where index i denotes contribution from i^{th} node density $\mathcal{S}_i$
- Extending the above method to whole $\mathcal{T}_\mathcal{G}$, we can compute the contribution of each node $\mathcal{S}_i$ to the whole graph. [Structure contribution by laplacian]
- At the end, each i^{th} node will have a vector (K) of $N_\mathcal{V} \times 1$, where K_j represents the contribution of j^{th} node towards i^{th} node.

Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

- Representation of subgraphs within molecule ?
- How to define spectral density?
- How to perform Fourier transform?
- How to define kernel?

How to define Kernel?

- Each node has a vector K of $N_{\mathcal{V}} \times 1$, and we define the kernel matrix for the node as KK^T which is a $N_{\mathcal{V}} \times N_{\mathcal{V}}$ providing the co-variance matrix for the G.P.

How to define Kernel?

- Each node has a vector K of $N_{\mathcal{V}} \times 1$, and we define the kernel matrix for the node as KK^T which is a $N_{\mathcal{V}} \times N_{\mathcal{V}}$ providing the co-variance matrix for the G.P.
- We define the gaussian process at each node by

$$\mathcal{F}_i \sim \mathcal{GP}(0, K_i K_i^T) \tag{6}$$

where i represents the node index. One can also define some non-linear variants of co-variance matrix by passing it through a NN or any non-linear activation function.

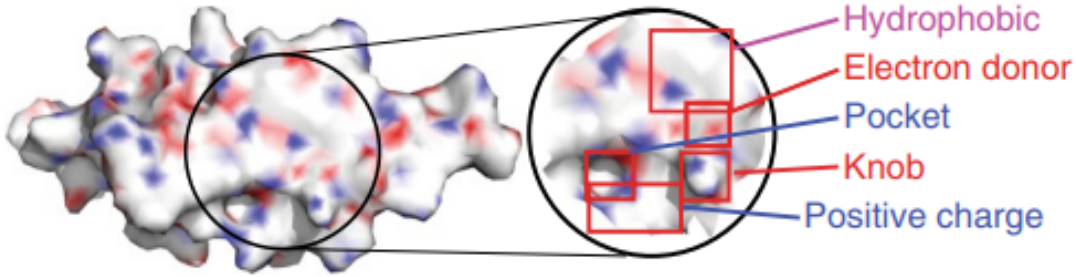
Spectral Density over Graphs

Aim: Define Spectral Density over subgraphs of a molecular graph

- Representation of subgraphs within molecule ?
- How to define spectral density?
- How to perform Fourier transform?
- How to define kernel?
- Where can i use it?

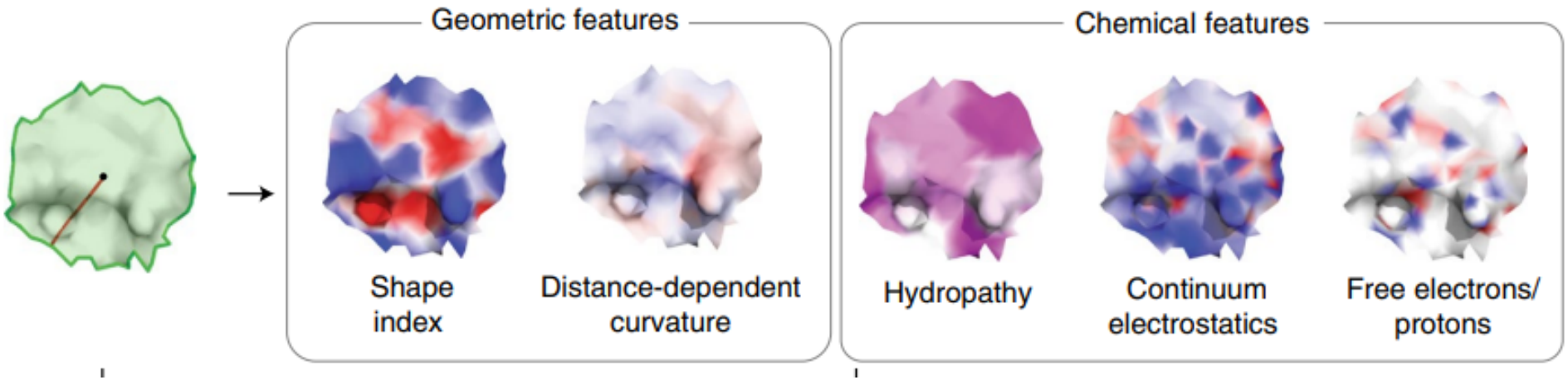
Where we can apply it?

- Manifold or mesh based representation of a molecule: Represented as a surface, where sub-regions will corresponds to certain spectral density [Maybe useful in drug target prediction]



Where we can apply it?

- Manifold or mesh based representation of a molecule: Represented as a surface, where sub-regions will corresponds to certain spectral density



Where we can apply it?

- Manifold or mesh based representation of a molecule: Represented as a surface, where sub-regions will corresponds to certain spectral density
- Can be applied in a generative model by providing a skeletal of molecule (junction tree) and inferring the true molecule or scoring sub-graphs for molecular addition
- Many more to think and explore!

Conformer generation by Z-matrix

Aim: Recover the conformer from a 2-D graph of a molecule

Literature Review

Method	C.C/D.G	Geometric Quantities	Physics
GEOMOL	✗	✓	✗
DGSM	✓	✗	✗
CGCF + ETM	✓	✗	✗[little bit]
GraphDG	✓	✗	✗
ChemNet + PhysNet	✓	✗	✓
??	??	??	??

Conformer generation by Z-matrix

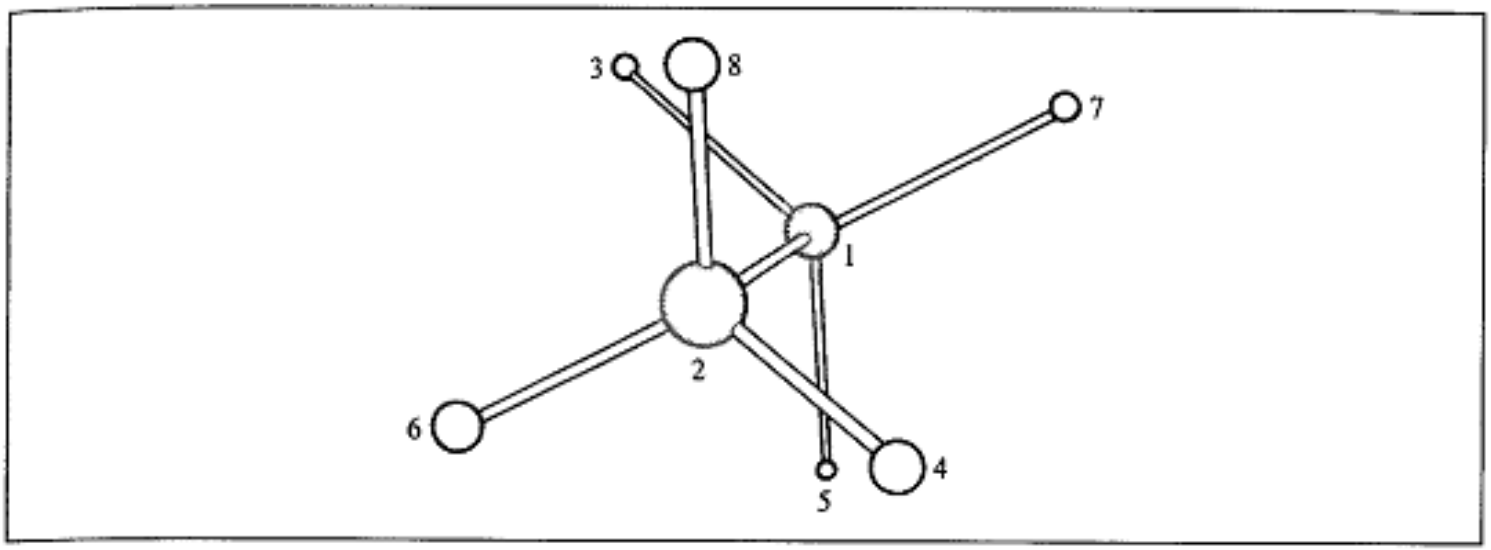
Aim: Recover the conformer from a 2-D graph of a molecule

- How to represent?

Z-matrix

- Represent a system built of atoms. A Z-matrix is also known as an internal coordinate representation.
- In terms of bond length, bond angle and dihedral angle; defined w.r.t to some atom in the molecule.
- Can be converted to Cartesian coordinates of the atoms of the molecule.

Z-matrix



as follows:

1	C						
2	C	1.54	1				
3	H	1.0	1	109.5	2		
4	H	1.0	2	109.5	1	180.0	3
5	H	1.0	1	109.5	2	60.0	4
6	H	1.0	2	109.5	1	−60.0	5
7	H	1.0	1	109.5	2	180.0	6
8	H	1.0	2	109.5	1	60.0	7

Conformer generation by Z-matrix

Aim: Recover the conformer from a 2-D graph of a molecule

- How to represent?
- Learning the mapping?

Learning the mapping?

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E, representing a molecule with nodes as atoms and edges as bonds

Learning the mapping?

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E, representing a molecule with nodes as atoms and edges as bonds
- Aim: Find the Z_{truth} given a 2-D molecular graph, then use it to compute Cartesian coordinates of the conformer (there can exist many Z_{truth}, as there are multiple conformer of single molecule)

Learning the mapping?

- One can neglect the bond length as $C - C$ *or* $C = C$*etc* length remain nearly same in all molecules, and we have the initial 2-D molecule graph, information about atoms, type, valency etc.
- Then it reduces to finding the angles (bond and dihedral) giving an orientation of how sub-molecules/atoms are arranged within a molecule.

Learning the mapping?

- Dynamics + inducing some Physics?

Dynamics

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph

Dynamics

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph
- Use bond length as edge feature and (bond angle, dihedral angle, mass, valency, atom type...etc) as node features

Dynamics

- Given a graph $\mathcal{G} = (V, E)$ where V are vertices (nodes) and E are edges of the graph
- Use bond length as edge feature and (bond angle, dihedral angle, mass, valency, atom type...etc) as node features
- Use a junction tree approach for updating of features if nodes and edges i.e.
  - Each of the corresponding nodes & edges in cluster will receive a message from all nodes & edges, within the cluster
  - Do the regular updates using AGNN or MPNN on junction tree.

Some Physics?

- Bond and Dihedral angles depend on positional and torsional forces within the molecule via interactions; depend on position and momenta

Some Physics?

- Bond and Dihedral angles depend on positional and torsional forces within the molecule via interactions; depend on position and momenta
- As we know bond angle, bond length and dihedral angle characterize internal coordinates of molecule $\rightarrow$ can be used to estimate (latently) positional and torsional forces

Forces

- An encoder can be used (similar to ODE^2VAE) to decompose into positional force ($f_{i,pos}$) and torsional force ($f_{i,tor}$), giving

$$q_{enc}(f_{i,pos}|\theta_1) \ \& \ q_{enc}(f_{i,tor}|\theta_1)$$

Forces

- An encoder can be used (similar to ODE^2VAE) to decompose into positional force ($f_{i,pos}$) and torsional force ($f_{i,tor}$), giving

$$q_{enc}(f_{i,pos}|\theta_1) \ \& \ q_{enc}(f_{i,tor}|\theta_1)$$

- Parameters θ_1 and θ_2 are the inputs to encoders, where $\theta_1 = [\theta_{1,1}, i \in \mathcal{N}(V)]$ corresponds to the positional force input parameters and $\theta_2 = [\theta_{2,1}, i \in \mathcal{N}(V)]$ corresponds to the torsional force input parameters defined as:

Forces

- An encoder can be used (similar to ODE^2VAE) to decompose into positional force ($f_{i,pos}$) and torsional force ($f_{i,tor}$), giving

$$q_{enc}(f_{i,pos}|\theta_1) \;\&\; q_{enc}(f_{i,tor}|\theta_1)$$

- Parameters θ_1 and θ_2 are the inputs to encoders, where $\theta_1 = [\theta_{1,1}, i \in \mathcal{N}(V)]$ corresponds to the positional force input parameters and $\theta_2 = [\theta_{2,1}, i \in \mathcal{N}(V)]$ corresponds to the torsional force input parameters defined as:

$$\theta_{1,i} = (\mathcal{F}_{V_i}(t)) \tag{7}$$

$$\theta_{2,i} = (\mathcal{F}_{V_i}(t-1:t), \bigoplus_{k}^{hops} \mathcal{F}_{E_{k,i}}(t-1:t), \mathcal{F}_{V_{i,j}}[\forall j \in 2\ hops](t-1:t)) \tag{8}$$

Forces

- An encoder can be used (similar to ODE^2VAE) to decompose into positional force ($f_{i,pos}$) and torsional force ($f_{i,tor}$), giving

$$q_{enc}(f_{i,pos}|\theta_1) \ \& \ q_{enc}(f_{i,tor}|\theta_1)$$

- Parameters θ_1 and θ_2 are the inputs to encoders, where $\theta_1 = [\theta_{1,1}, i \in \mathcal{N}(V)]$ corresponds to the positional force input parameters and $\theta_2 = [\theta_{2,1}, i \in \mathcal{N}(V)]$ corresponds to the torsional force input parameters defined as:

$$\theta_{1,i} = (\mathcal{F}_{V_i}(t)) \tag{7}$$

$$\theta_{2,i} = (\mathcal{F}_{V_i}(t-1:t), \bigoplus_{k}^{hops} \mathcal{F}_{E_{k,i}}(t-1:t), \mathcal{F}_{V_{i,j}}[\forall j \in 2 \ hops](t-1:t)) \tag{8}$$

- Alternative: Use a similar ODE formalism as in ODE^2VAE

Forces

- These $f_{i,pos}$ & $f_{i,tor}$ can be used as an additional parameter when computing the update for the node and edge features.
- The positional force can be used as it is, whereas the torsional force can be used as weighted sum given as

$$F_{i,tor} = \sum_{k}^{hops} (B.L)_{ik} f_{k,tor} \tag{9}$$

Forces

- These $f_{i,pos}$ & $f_{i,tor}$ can be used as an additional parameter when computing the update for the node and edge features.
- The positional force can be used as it is, whereas the torsional force can be used as weighted sum given as

$$F_{i,tor} = \sum_{k}^{hops} (B.L)_{ik} f_{k,tor} \tag{9}$$

- The $F_{i,tor}$ represents the total torsional force by other neighbouring atoms (intuitive as dihedral angle is defined w.r.t neighbouring atomic plane)
- We can change to weighting factor to any other factor, but bond length seems natural choice

Learning the mapping?

- Dynamics + inducing some Physics?

Learning the mapping?

- Dynamics + inducing some Physics?
- Angles as Von Mises distribution

Von Mises distribution

- Von Mises distribution is a continuous probability distribution on the circle deemed as circular analogue of the normal distribution.

$$f(x \mid \mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{10}$$

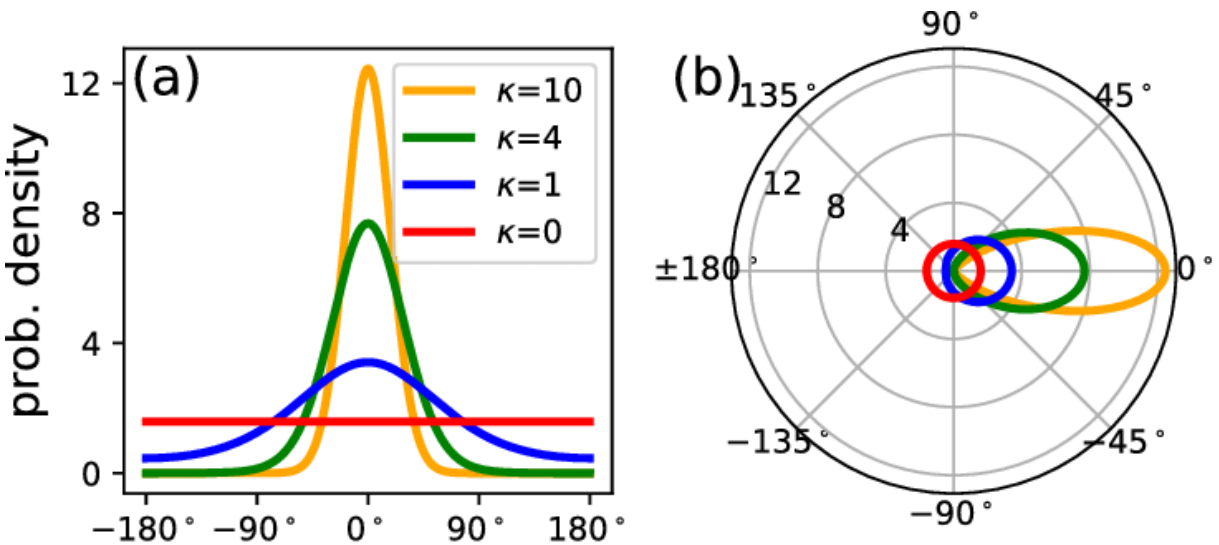
where $I_0(\kappa)$ is the modified Bessel function of order 0.

Von Mises distribution

- Von Mises distribution is a continuous probability distribution on the circle deemed as circular analogue of the normal distribution.

$$f(x \mid \mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{10}$$

where $I_0(\kappa)$ is the modified Bessel function of order 0.



Learning the mapping?

- Angles can be estimated as MLE of Von Mises distribution
- Reducing to

$$max \sum_{all\ \ angles} cos(\angle Pred - \angle Truth) \tag{11}$$

Conformer generation by Z-matrix

Aim: Recover the conformer from a 2-D graph of a molecule

- How to represent?
- Learning the mapping?
- Predicting the Cartesian coordinates from Z matrix

Literature Review

Method	C.C/D.G	Geometric Quantities	Physics
GEOMOL	✗	✓	✗
DGSM	✓	✗	✗
CGCF + ETM	✓	✗	✗[little bit]
GraphDG	✓	✗	✗
ChemNet + PhysNet	✓	✗	✓
GPMG	✗	✓	✓

THANK YOU