😀

[MT 발표]Generative Coarse-Graining of Molecular Conformations 20220926

Tags	☐
Day	2022.09.26
Property	
Topic	GNN Generative Molecular
Who	DJ
Year, Journal, Conference	2022 ICML
code	
comment	
url	https://arxiv.org/pdf/2201.12176.pdf

Contents

0. Background

> 궁금하시면 하이퍼 링크를 눌러보세요

- Molecular simulations
- Molecular Modeling : blog1, Bric

▼ 정리

분자모델링 : 화학적 생물학적 실험으로 하기 어렵거나, 너무 많은 시간이 걸릴경우 컴퓨터로 시뮬레이션을 하는 것을 말한다.

1) Quantum Mechanics : 작은 분자에 해당

2) Molecular Dynamics : 큰 분자를 이용하면 전이상태의 구조를 간접적으로 확인 가능

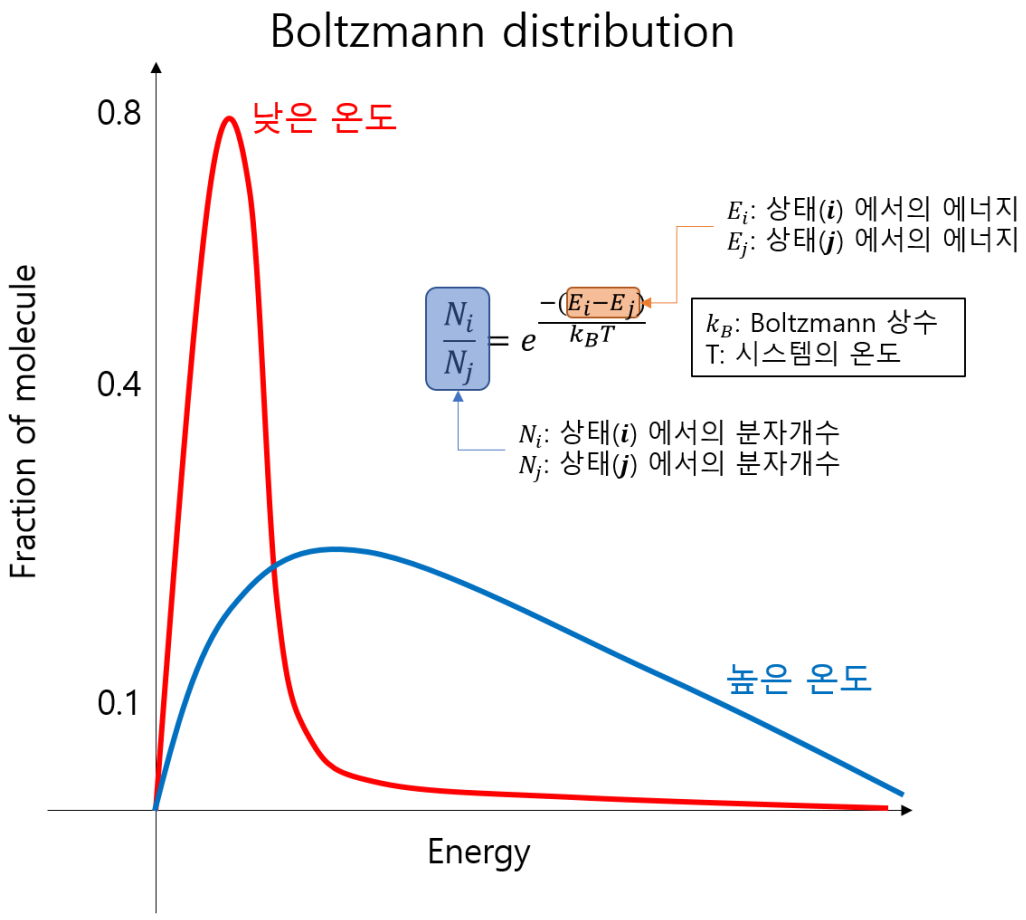
- Molecular dynamics : youtude, paper

▼ 정리

많은 분자들로 구성된 시스템의 성질은 구성 분자들의 서로 다른 구조 및 에너지 상태의 종합적인 결과라고 할 수 있다. 단백질-리간드 복합체를 하나의 시스템으로 본다면 많은 원자들로 구성된 시스템이라고 할 수 있으며, 구성 원자들의 위치와 움직임에 따라 구조 및 상호작용의 변화가 일어나는 동역학적인 시스템이라 할 수 있다. 이러한 상호작용을

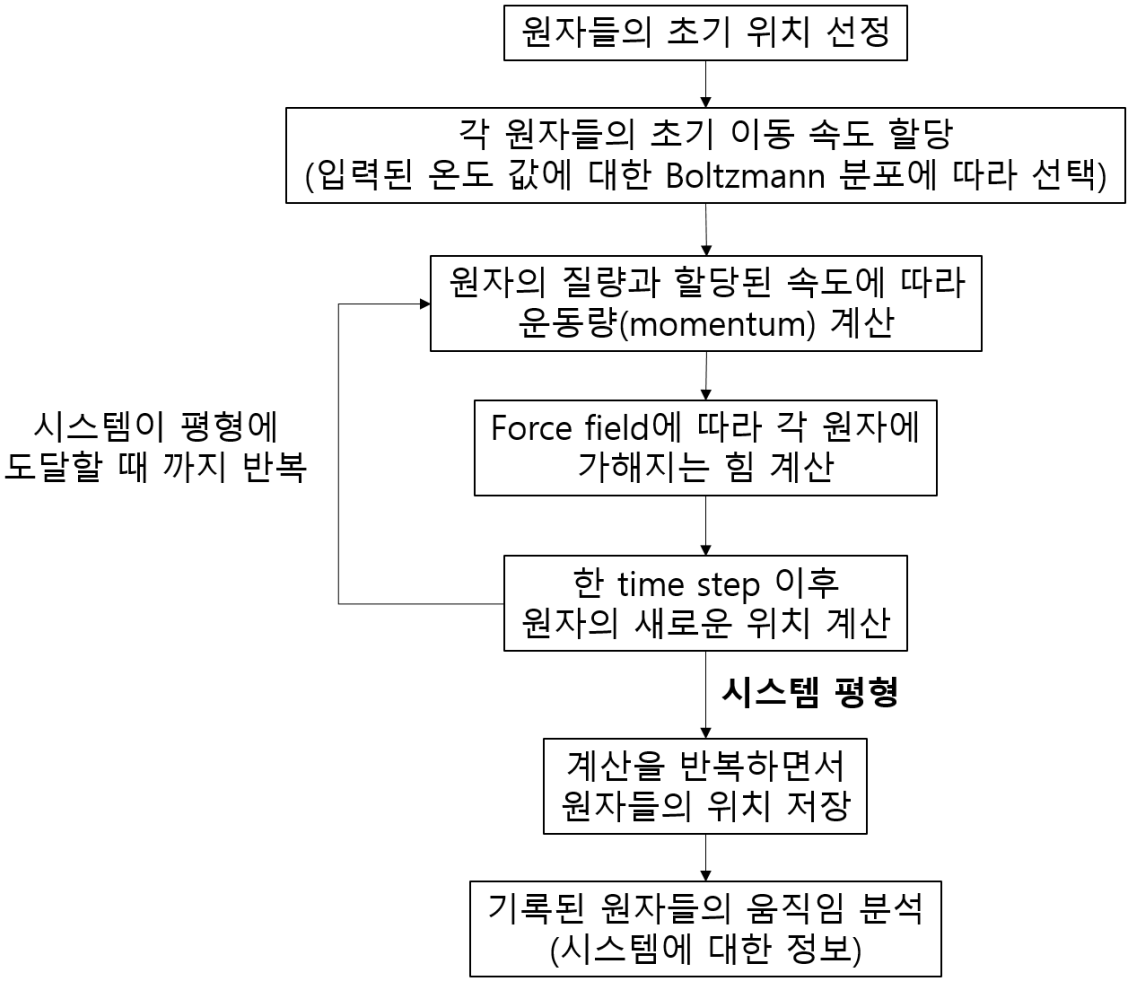
시뮬레이션하기 위한 방법으로 molecular dynamics (MD)와 Monte Carlo (MC) simulation이 있다.

1. boltzmann distribution : system전체의 에너지는 system의 온도를 통해 측정이 가능하지만, system전체의 에너지가 system을 구성하는 각 분자, 원자의 개별적인 에너지라고 할 수 없음
   ⇒ 분자 및 원자들의 서로 다른 에너지 값들은 어떤 확률 값에 따라 분포되어 있는데, 이 분포를 Boltzmann 분포라 하며, system의 온도에 의존적으로 확률값이 변화하게 됨



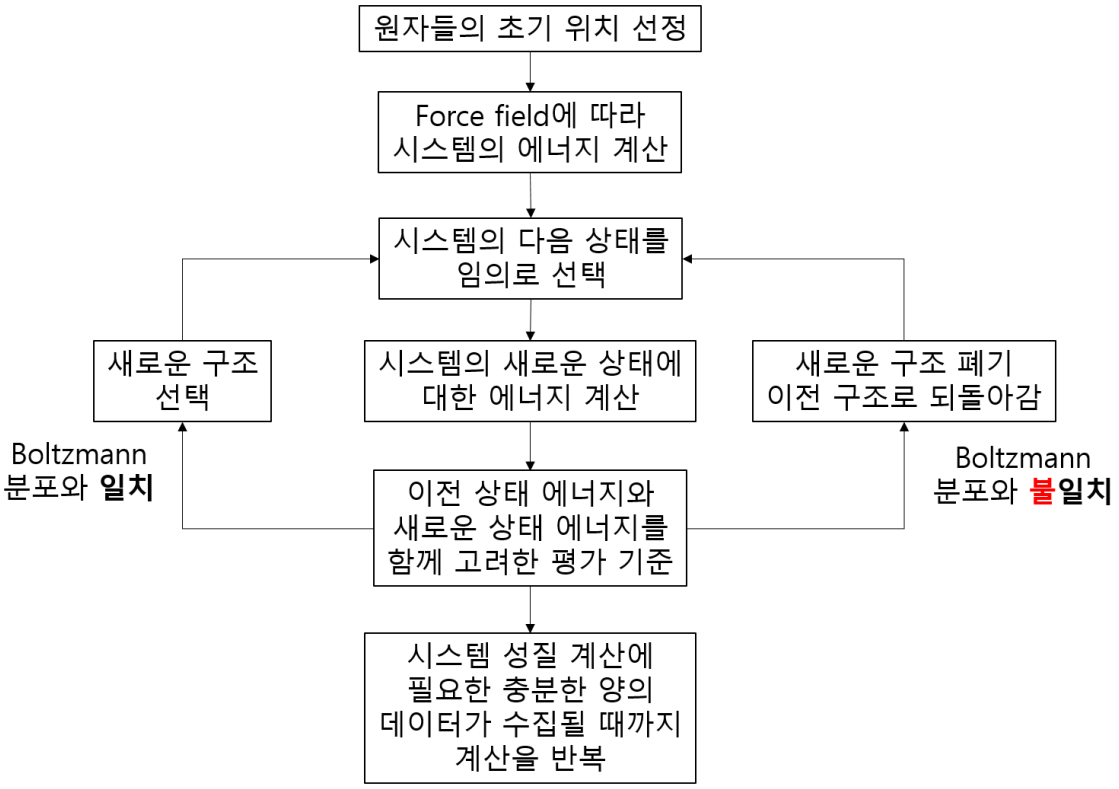
2. Molecular dynamics

⇒ 시간에 따른 시스템의 변화를 시뮬레이션 하는 방법으로 Force filed에 기반한 분자역학에 따라 시스템의 에너지를 계산하게 되는데, 시스템을 구성하는 각 원자들에 가해지는 힘을 계산해서 구조의 변화와 그 구조에서의 에너지 값을 계산하게 된다.



3. Monte Carlo

→ 난수값을 이용한 random sampling방법으로, 분자 모델링에서의 MC는 분자 구조를 통계적인 분포에 따라 임의적으로 바꾸면서 시뮬레이션을 진행하게 된다.



- Molecular docking

▼ 정리

⇒ molecular docking은 atom-level에서 ligand와 단백질간의 상호작용을 모델링하여 리간드-단백질 결합 구조 (ligand-protein complex)를 예측하는 것이 목표

⇒ 이를 통해 표적 단백질의 결합 부위(binding site)내에서 ligand가 어떻게 행동하는지 규명가능

⇒ 단백질의 활성 부위 (active site)와 ligand간의 결합 가능한 경우의 수가 너무 많기 때문에 모든 ligand conformation을 만들어 비교하는 건 불가능

sampling 알고리즘

1. Pharmacophore 기반 알고리즘 : 분자의 형태와 화학적 정보를 기반으로 매칭, 단백질의 active site와 ligand를 pharmacophore로 표현해 각 분자별 pharmacophore간의 거리를 기반으로 매칭이 이루어짐
2. Fragment 기반 알고리즘 : ligand를 몇몇의 fragment들로 나누고 각각의 조각들을 docking하는 방법
3. Stochastic 탐색 알고리즘 : ligand conformation을 임의로 수정하면서 가능한 conformation을 찾아가는 방법

- Coarse-graining : Paul

- Conformation

▼ 정리

⇒ 입체구조, 어느 사물이 속한 집단이나 정체성이 무엇인지를 식별해 주는 입체적인 모양 내지 구조 [단순한 shape, structure, form으로 쓰지 않는 이유]

- VAE

1. Abstract & Introduction

Coarse-graining란?

> 💡 ***분자 시뮬레이션의 Coarse-graining(CG)는 선택된 원자를 pseudo-beasds grouping하여 particle represenation(입자표현)을 단순화하고, 시뮬레이션을 가속화***한다.

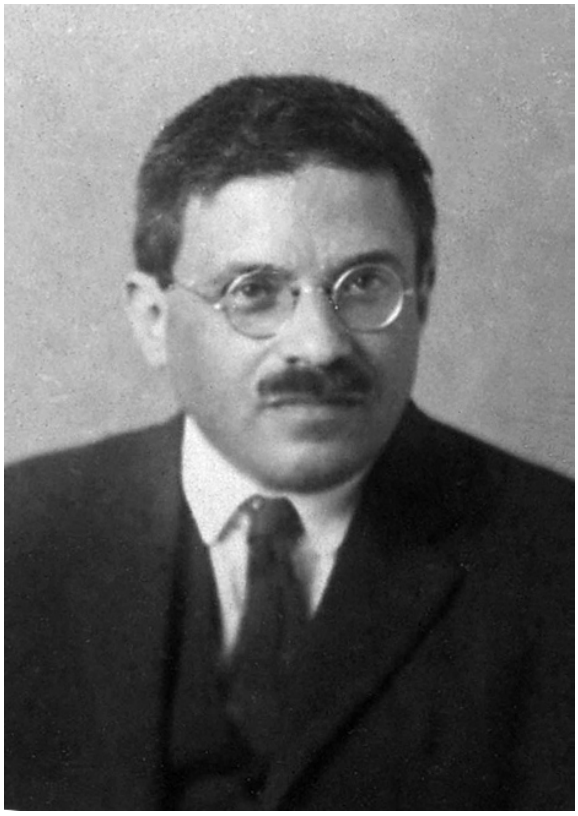
- 하지만, CG과정은 정보 손실을 가속화 한다.

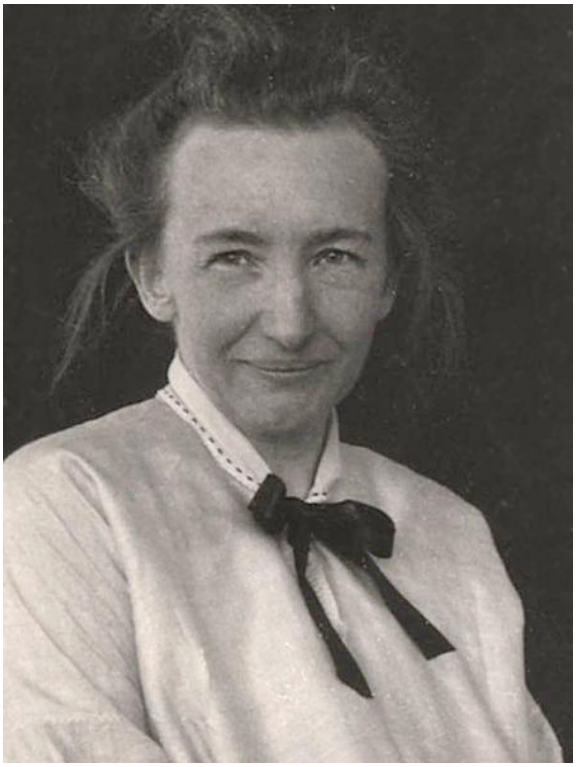
⇒ CG좌표에서 fine grained coordinates로 복원하는 정확한 back mapping이 어렵기 때문에, 이는 challenging한 문제임 [분자 모델링과 관련있을듯]

- 논문에서는 Generative model, Equivariant network에서 영감을 받아, backmapping transformaion의 필수 조건인 **확률적 성질(latent space, learning probability)**과, **geometric consistency(Equivariant)**를 포함하여 엄격하게 embeds하는 model제안

⇒ FG를 latent space로 encoding하고 equivarint network decoder를 통해 다시 디코딩을 한다.

- **파울 에렌페스트 & 타티아나 아파나시예바[부부십니다] : 신동희 선생님 좋아하는거 다 압니다.**





⇒ 이 두분이 continuous distribution위에서 차원 축소를 하는 방법인 **Coarse-graining**을 제안함

- 물리학에서 복잡한 문제를 단순화하는 powerful한 tool!
- ***Molecular simulation***에서 CG는 rule-based CG mapping을 통해 original atoms x를 개별적인 beads X로 lumping groups을 통해 particle representation의 단순화를 위해 사용되었다
- CG molecular dynamics(MD)는 chemical spaces에서 단순한 combination rules과 함께 상당한 계산량 speed up을 할 수 있는 방법임

⇒ large molecules with hundreds of thousands of atoms sucah as biological and artificial polymers

⇒ 분자 시뮬레이션 방법론은 크게 두 가지로 구분 됨

1. Monte Carlo 방법론(확률론 기반, rejection, acception비율..)
2. Molecular dynamics방법론(시간에 따른 분자의 운동 궤적을 구하고, 이를 바탕으로 시스템의 동역학 성질 계산)

- CG MD는 protein folding이나 polymer reptation같은 long times of phenomena에 접근되도록 허락된 기술..? ⇒ 오랫동안 저 두 분야에 쓰였다는건지..?

> 하지만 이러한 accelation은 fine-grained atomic detail의 정보 손실을 가져옴

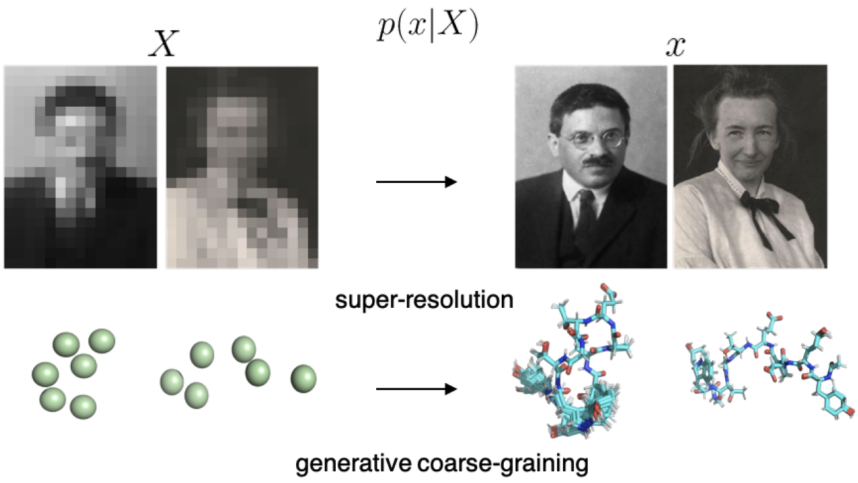


Figure 1: An analogical illustration of the tackled generative coarse-graining problem. **Top**: Image super-resolution recovers higher resolution portraits[1]. **Bottom**: Generative CG generates fine-grained molecular structures.

- **CG**는 super-resolution과 비슷하게 생각할 수 있는데 [논문에서 의도한건지는 모르겠음], molecular simulations에 있어 입자표현을 pseudo-beads로 간단히 표현해서 drastically한 accelerates를 보이는 방법으로 사용 중

저자들이 해결하고자 하는 문제!

💡 어떻게 하면 CG coordinate X에서 FG 구조 x를 정확히 recover할 수 있을까?

1. backmapping problem based on random projection approach

한계점

1) poor initial geometries를 가지고 있고

2) specific fragment libraries도 필요하며

3) non-data driven으로 predefined fragment geometries에 편향됨

2. Parameterized functions to deterministically backmap approach

⇒ data-informed backmapping solutions with machine learning

한계점

1) low-quality geometries

2) complex한 molecular structure에는 test를 하지 않음

Challenging

C1 : Stochasticity of backmapping

⇒ 너무 많은 FC configuration이 같은 CG conformation으로 projections : **one-to-many(reverse)**

⇒ 다양한 구조를 만들기 어렵다.

C2 : Geomety consistency

⇒ geometric consistency constraints는 기존 연구들에서 고려되지 않았다 : **Equivariant**

C3 : Generality w.r.t mapping protocols and resolutions

⇒ FG에서 CG로 가는 general한 protocol이 존재하지 않는다.

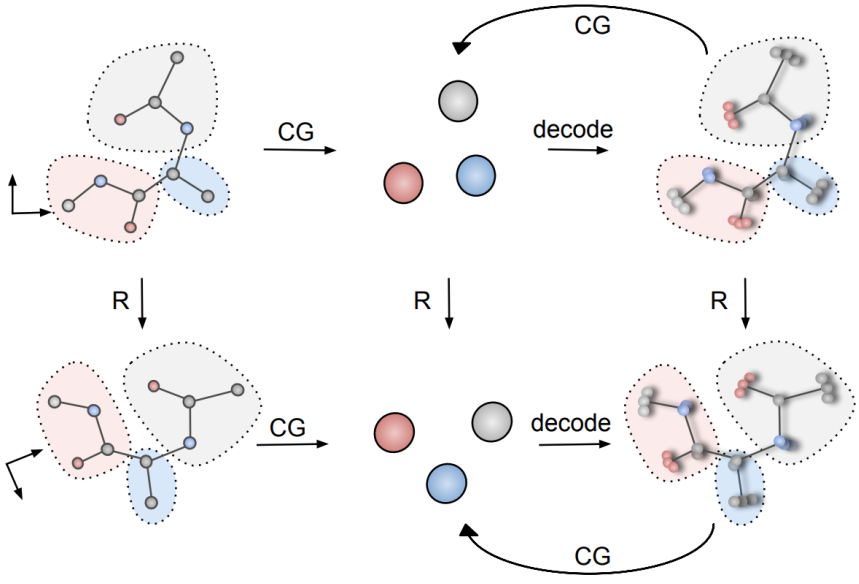


Figure 2: A diagram illustrating the geometric constraints for the backmapping function: 1) The backmapped coordinates should map back to the original CG coordinates; 2) because the CG transformation is equivariant, the backmapping transformation should also be equivariant.

Contribution

💡 Coarse-Graining Variational Auto-Encoder (CGVAE)를 제안
→ 위에 있는 challenges들과 novel probabilistic인 모델
→ stochastic backmapping문제를 conditional generative task로 변환
$p_\theta(x|X)$: CG(X)가 주어질때, FG(x)를 recover

1. [C1]molecular conformation(분자입체구조)의 backmapping problem을 위해 probabilistic formulation원칙을 제공
   + CGVAE : latent variable model
2. [C2]rigorous하고 expressive한 backmapping function을 위해 equivariant convolution을 design
3. [C3] Molecular dynamics simulations에서 diverse mapping protocol로 응용가능한 agnostic한 model 제안
4. two benchmark dataset 제안

2. Related works

1) Backmapping projection

2) Parameterized deterministic function

- 위 두 개의 방법은 lack equivariance and chemical ruile supervision
- 그리고 backmapped system의 intensive force field equilibration에 의존

💡 현재 C1-C3를 고려한 연구는 없다고 주장함

3. Preliminaries

CG Representations of Molecular Structures

💡 - fine-grained(FG) system : $x = \{x_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$
- coarse-grained(CG) sysyem : $X = \{X_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$
- CG operation : m : [n] → [N]
- FG atom i, CG atom I
- $C_I = (k \in [n] | m(k) = I)$
- CG projection operation $X = Mx, M \in \mathbb{R}^{N\times n}$ with $M_{I,i} = \frac{w_i}{\sum_{j\in C_I} w_j}$

Geometric Requirements of Backmapping

Property 3.1. *Let* $f_M : \mathbb{R}^{n\times 3} \to \mathbb{R}^{N\times 3}$ *be the linear transformation* $f_M(x) := Mx$. f_M *is* $E(3)$ *equivariant, i.e.,* $f_M(Qx + g) = Qf_M(x) + g$, *where* Q *is a* 3×3 *orthogonal matrix, and* g *is a translational vector.*[2]

R1. $M\widetilde{x} = M\mathrm{Dec}(X) = X$.
R2. $\mathrm{Dec}(QX + g) = Q\mathrm{Dec}(X) + g$.

- R1 : self-consistency, MDec(X)=X
- R2 : decoder equivariant

4. Method

Overview

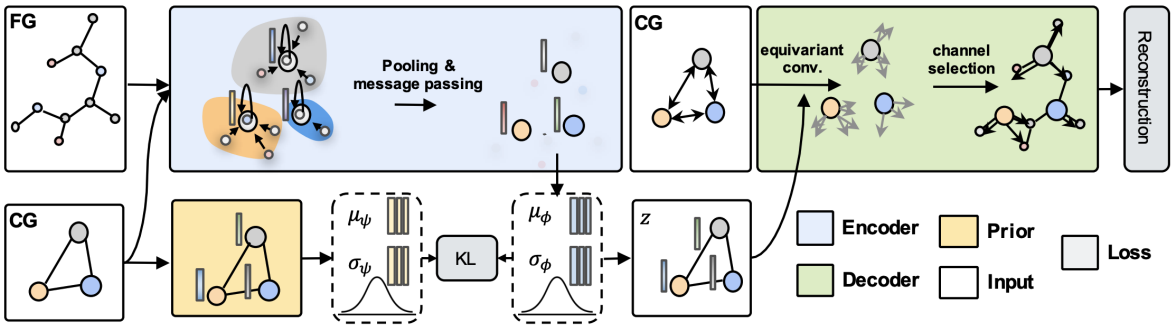


Figure 3: The overall framework of generative CG modeling. The encoder $q_\phi(z|x, X)$ takes both FG and CG structures as inputs, and outputs invariant latent distributions for each CG node via message passing and pooling. Given sampled latent variables and the CG structure, the decoder $p_\theta(x|X, z)$ learns to recover the FG structure through equivariant convolutions. The whole model can be learned end-to-end by optimizing the KL divergence of latent distributions and reconstruction error of generated FG structures.

4.1 Generative Coarse-Graining Framework

- we study the inverse problem of recovering x from X : CG→FG

> 💡 Propose to learn a parameterized conditional generative model $p_\theta(x|X)$ to approximate the recovering function

- $p(x|X) = \int p_\theta(x|X, z)p_\psi(z|X)dz$ → intractable

⇒ Variantional inference 적용!

$$\log p(x|X) \geq \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log p_\theta(x|X, z)}_{\mathcal{L}_{\text{recon.}}} + \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log \frac{p_\psi(z|X)}{q_\phi(z|x, X)}}_{\mathcal{L}_{\text{reg.}}} \quad (1)$$

ELBO

▼ VAE 참고

ELBO

$$\log p_\theta(x^{(i)}) = E_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right]$$

$$= E_z\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}\right] \quad \text{Bayes' Rule}$$

$$= E_z\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}\frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}\right]$$

$$= E_z\left[\log p_\theta(x^{(i)}|z)\right] - E_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)}\right] + E_z\left[\log \frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}\right] \quad \text{KL divergence}$$

$$= E_z\left[\log p_\theta(x^{(i)}|z)\right] - D_{KL}(q_\phi(z|x^{(i)})||\, p_\theta(z)) + D_{KL}(q_\phi(z|x^{(i)})||\, p_\theta(z|x^{(i)}))$$

*참고

Expectation $E_{p(x)}[f(x)] = \int f(x)p(x)dx$

KL-divergence $KL(P||Q) = \sum P(x)\log\frac{P(x)}{Q(x)}$

Model

- $p_\theta(x|X,z)$: decoder
- $q_\phi(x|X,z)$: encoder
- $p_\psi(x|X)$: prior model ⇒ respectively to model the uncertainties from CG reduction

Loss

> L_{recon} : the expected reconstruction error of generated FG structures
> L_{reg} : a regularization over the latent space

Encoder & Prior

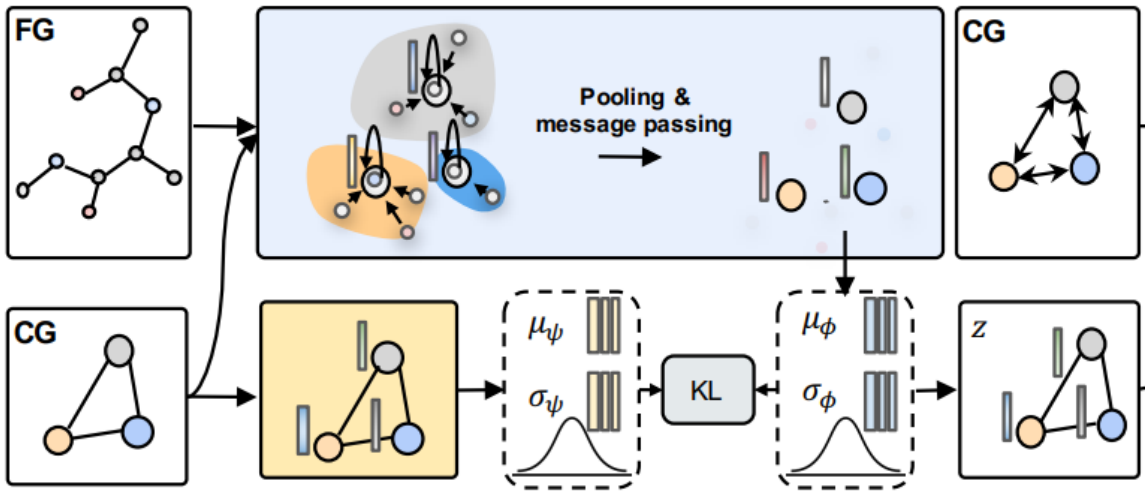
4.2 CG Encoding and Prior

Overview of the data representation

⇒ Encoder는 FC structure, CG structure둘 다 input으로 받아 정보를 추출

⇒ node label은 atom type, edge는 labeled with distance로 구성

Estimating the posterior distribution



⇒ (X, x) → z

> 💡 Encoder는 three operation으로 CG-level의 invariant embedding 추출한다.

step 1 : message passing at FG level

- MPNN 구조

$$h_i^{t+1} = \text{Update}\Big(\sum_{j \in N(i)} \text{Msg}(h_i^t, h_j^t, \text{RBF}(d_{ij})),\ h_i^t \Big), \qquad (2)$$

→ 기본적으로 MPNN을 따라고, edge feature는 Radial basis transformation을 사용해서 high dimension feature로 translation

→ initial node embedding은 atomic types

step 2 : pooling operation that maps FG space to CG space

$$\widetilde{H}_I^t = \text{Update}\Big(\sum_{i \in C_I} \text{Msg}\left(h_i^{t+1}, H_I^t, \text{RBF}(d_{iI})\right),\ H_I^t \Big), \qquad (3)$$

→ pooling을 통해 FG embedding을 CG-level로 변환

→ h_i^{t+1} : FG embedding

→ H_I^0 : initialized 0_F

→ FG node로부터 할당된 CG node로 pools message

step 3 : message passing at CG level

$$H_I^{t+1} = \text{Update}\left(\sum_{J \in N(I)} \text{Msg}(\tilde{H}_I^t, \tilde{H}_J^t, \text{RBF}(d_{IJ})),\ H_I^t \right), \quad (4)$$

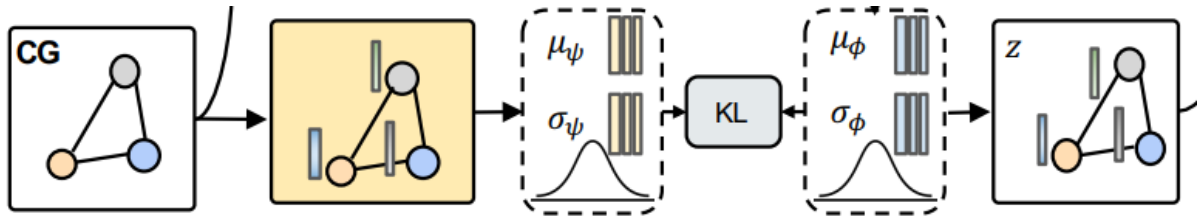
→ CG-level에서 MP진행

Final embedding

$$\text{multivariate Gaussians } \mathcal{N}(z_I | \mu_\phi(H_I^\phi), \sigma_\phi(H_I^\phi)).$$

→ two feedforward networks으로 u, sigma를 modeling

Estimating the prior distribution



💡 X→z, prior model은 CG structure를 받아서 latent space으로 embedding

→ initial CG node features는

$$H_I^0 = \sum_{i \in C_I} h_i^0.$$

the pooled sum of one-hot FG fingerprints

→ Eq(4)와 동일한 신경망 구조로 구성되어있고, final CG embeddings Prior H에서 두 개의 seperate NN u, sigma를 통해 normal distribution의 parameterized를 해준다.

Decoder

4.3 Multi-channel Equivariant Decoding

❓ 궁금한점… : 어떤 부분에서 정확히 equivariant한 decoding이 되는지는, PaiNN논문을 봐야할 수 있을듯
ref. 1 : Equivariant message passing for the prediction of tensorial properties and molecular spectra

💡 논문의 Decoder design은 vector-based graph neural nets에서 영감받았다고 함
ref.1 : Equivariant message passing for the prediction of tensorial properties and molecular spectral 2021
ref.2 : E(n) equivariant graph neural networks, 2021
ref.3 : Learning from protein structure with geometric vector perceptrons, 2020
ref.4 : Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials

💡 **Decoder : $p_\theta(\tilde{x}|X,z)$: CG X와 latent z를 받고 FG geometry x를 예측**

Three step :

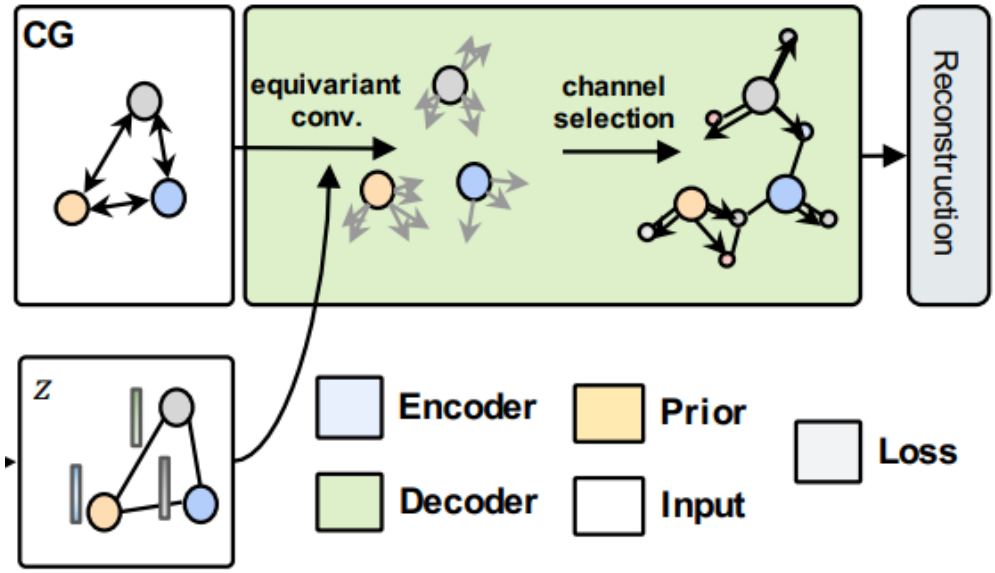
1) Given X and invariant feature z from the encoder

⇒ generate F equivariant vector channel for each bead : $V^\theta \in \mathbb{R}^{N \times F \times 3}$

2) channel selection : Each bead corresponds to different number of atoms, not all channels will be used

3) relative position to absolute : satisfy **R1 [**$MDec(x) = X$**]**

$\triangle\tilde{x} \rightarrow \tilde{x}$



1) Equivariant convolution

$\Rightarrow (X, z) \rightarrow V^{\theta}$

: using equivariant convolutions based on inter-node vectors to predict $\triangle\tilde{x}$

MP operation
- node-wise scalar feature$H \in \mathbb{R}^F$
- pseudoscalar features $\bar{H} \in \mathbb{R}^F$
- vector features $V \in \mathbb{R}^{F\times 3}$
- pseudovector features $\bar{V} \in \mathbb{R}^{F\times 3}$

⇒ encoder와 달리 set of edge-wise equivariant vector features를 input으로 받는다.

$$\{\hat{E}_{IJ} = \frac{X_J - X_I}{d_{IJ}} \mid (I, J) \in \mathcal{E}_{\mathrm{CG}}\}$$

⇒ chirality의 추가 정보를 포함하기 위해 pseudoscalars/pseudovectors를 포함한다.

> Chirality : 분자내 대칭인 면이 없으며, 거울상을 서로 겹칠 수 없는 분자

- convolutional updates in decoder

$$
\begin{aligned}
\Delta H_I^t &= \sum_{J \in N(i)} W_1 \circ H_J^t \\
\Delta \overline{H}_I^t &= \sum_{J \in N(i)} V_I^t \cdot \overline{V}_J^t \\
\Delta V_I^t &= \sum_{J \in N(i)} \Big(W_2 \circ (V_I^t \times \overline{V}_J^t) + W_3 \circ \overline{H}_I^t \circ \overline{V}_J^t \\
&\qquad + W_4 \widehat{E}_{IJ} + W_5 \circ V_J^t \Big) \\
\Delta \overline{V}_I^t &= \sum_{J \in N(i)} \Big(W_6 \circ (V_I^t \times V_J^t) + W_7 \circ (\overline{V}_I^t \times \overline{V}_J^t) \\
&\qquad + W_8 \circ \overline{V}_J^t + W_9 \circ \overline{H}^t \circ V_J^t \Big)
\end{aligned}
\tag{5}
$$

x : cross product
o : element-wise
' : dot product

→ initialize :

$$\overline{H}_I^0 = 0_F, V_I^0 = 0_{F \times 3}, \text{ and } \overline{V}_I^0 = 0_{F \times 3}.$$

⇒ H_I^0 는 training을 위해 $z \sim q_\phi(z|X, x)$에서 얻어지고 sampling을 위해 $p_\psi(z|X)$로부터 구한다.

> ❓ detail한 작동 원리는 appendix를 참고해야함

▼ detail한 작동원리

$W_1 - W_9$ 은 invariant edge-wise filters
$L_1 : \mathbb{R}^F \to \mathbb{R}^F$

$L_2 : \mathbb{R}^K \rightarrow \mathbb{R}^F$

- each filter implementation

$$W \in \mathbb{R}^F = L_1(\mathrm{RBF}(d_{IJ})) \circ L_2(H_J).$$

- Message passing operation은 equivariant update block $\sigma(\cdot, \cdot)$: PaiNN

⇒ update block : linearly mixed vector channels and introduces non-linear coupling between H and V

- For pseudoscalar and pseudovector

⇒ residual update to ensure that they flip sign under reflection

$$\begin{aligned} H_I^{t+1}, V_I^{t+1} &= \sigma(H_I^t + \Delta H_I^t, V_I^t + \Delta V_I^t) \\ \overline{H}_I^{t+1}, \overline{V}_I^{t+1} &= \overline{H}_I^t + \Delta \overline{H}_I^t, \overline{V}_I^t + \Delta \overline{V}_I^t \end{aligned} \tag{6}$$

▼ 이해안가는 문단

Pseudoscalars and pseudovectors in our decoder come from cross product updates. This introduces a richer basis set for the coordinate construction especially for low N cases. When N = 3, the span of the vector basis will be constrained in a plane, and cross product can overcome this limitation by introducing a vector basis in the orthogonal directions, increasing the expressiveness of the model.

⇒ N=3일때, vector basis span이 plane에서 제한되는데, cross product는 orthogonal direction으로 vector basis를 도입하여 모델의 표현력을 증가시켜 이 한계를 극복할 수 있다.

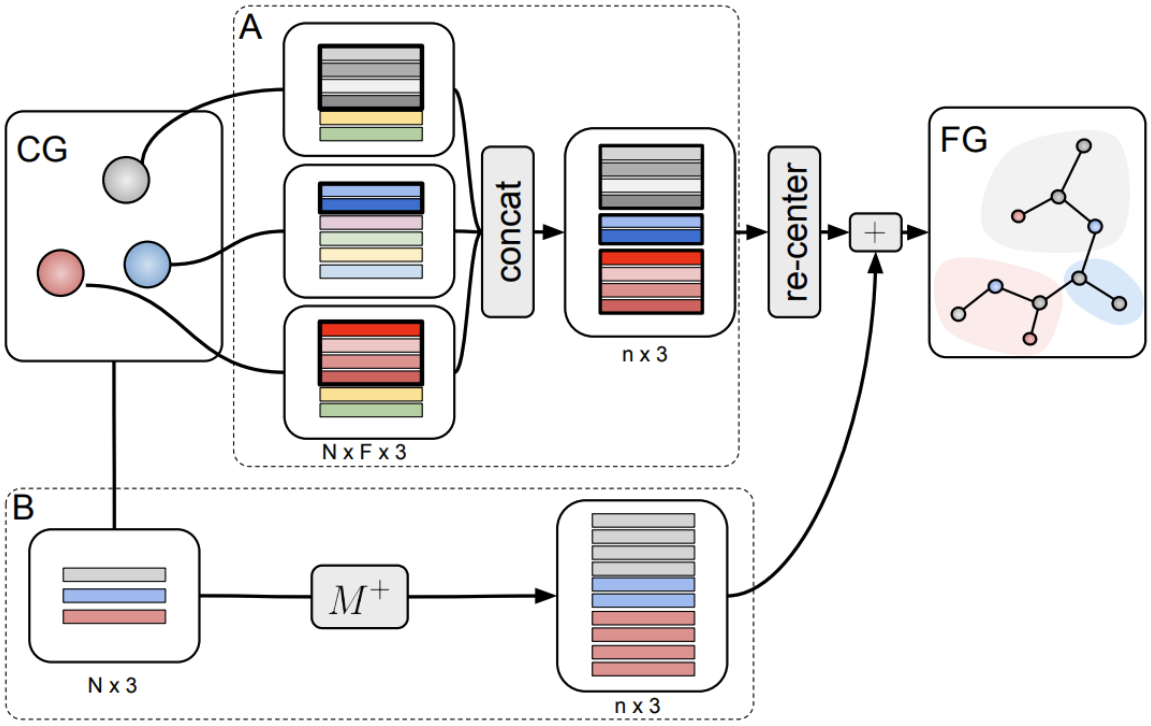
결론

💡 convolution update후 decoder는 final vector output $V^{\theta} \in \mathbb{R}^{N \times F \times 3}$을 내보낸다.
→ V^{θ} 가 E(3) equivariant 함을 보이는건 쉽다고 함 : strictly with vector operation으로 update를 했기때문
→ R2 만족 [Equivariant]

2) Channel selection

❓ 이 과정이 왜 필요한지? : equivariant convolution update와 관련이 있을거 같은데, PaiNN을 보고 확인해야함

$V^{\theta} \rightarrow \triangle \tilde{x}$



A : The prediction of $\Delta \tilde{x}_i$ are compiled from selected vector channels of CG node $m(i)$ based on their index in set C_I

B : lifting operator M^+로 M^+X통해서 CG coordinates에서 FG space로 lifted

💡 how to obtain the relative position predictions $\triangle\tilde{x} \in \mathbb{R}^{n\times 3}$ from the equivariant vector outputs $V^{\theta} \in \mathbb{R}^{N\times F\times 3}$

- 각 원자 i는 $I = m(i)$에 assigned됨에 따라, position $\triangle\tilde{x}$ 의 예측은 C_I에서 i의 index를 갖는 vector channel로 selection된다.

⇒ $Index(i, C_I)$: ex) Index(1, (1,2,4)) = 0

- V^{θ}로부터 channel selection후 FG-level prediction $\triangle\tilde{x}$ form으로 concatenate된다.

$$\Delta\tilde{x} = \bigoplus_i V^{\theta}_{m(i),\text{Index}(i,C_I)}$$

⇒ $m(i)$: selects CG beads, $Index(i, C_I)$: selects vector channel of bead I

▼ code

```
def channel_select(V, m):
    # m: n x 1  CG map for each FG node
    # V : N x F x 3  vector channels on CG node
    channel_idx = torch.zeros_like( m)
    for cg_type in torch.unique(m):
        cg_filter = m == cg_type
        # find size of C_I
        k = cg_filter.sum().item() # find size of C_I
        # construct (I, Index(i, C_I))
        channel_idx[cg_filter] = torch.LongTensor(list(range(k)))
    dx = V[m, channel_idx]
    return dx
```

3) Complie predictions for FG coordinates

💡 $\triangle\tilde{x}$은 relative position이기 때문에, final FG coordinates $\tilde{x}$ 으로 one step을 더 가야한다.

$\tilde{x} := M^{+}X + \Delta\tilde{x} - M^{+}M\Delta\tilde{x}$

$M^{+} \in \mathbb{R}^{n\times N}$

⇒[R1만족] M^{+} maps the point sets space from $\mathbb{R}^{N\times 3}$ to $\mathbb{R}^{n\times 3}$ by assigning or “lifting” CG coordinates back to their contributing FG atoms in C_I . The term $-M^{+}M\Delta\tilde{x}$ is added to re-center $\Delta\tilde{x}$ so that we can get the original X back after CG projection in order to satisfy **R1**

4.4 Model Training and Sampling

- Reconstruction loss calculation

$$\mathcal{L}_{\text{MSD}} = \frac{1}{n}\sum_{i=1}^{n} \|\widetilde{x}_i - x_i\|_2^2$$

$$\mathcal{L}_{\text{graph}} = \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} (d(\widetilde{x}_i, \widetilde{x}_j) - d(x_i, x_j))^2,$$

⇒ L_{graph}는 valid한 edge distance가 생성되도록하는 loss term

⇒ 생성된 분자 그래프에서 edge는 화학결합의 set을 나타냄

⇒ multi hop edge를 사용 [two-hop]

최종 recon- loss term

$$\mathcal{L}_{\text{recon.}} = \mathcal{L}_{\text{MSD}} + \gamma \mathcal{L}_{\text{graph}},$$

- Training and sampling

$$\mathcal{L} = \mathcal{L}_{\text{recon.}} + \beta \mathcal{L}_{\text{reg.}}$$

⇒ during training : CG latent variable z from

$$q_\phi(z|x, X) \text{ by } z = \mu_\phi + \sigma_\phi \circ \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I).$$

⇒ for sampling : the invariant latent variable

$$z \sim p_\psi(z|X)$$

⇒ finally, (input : z, X, output : x)

$$\widetilde{x} = \mathbf{Dec}_\theta(X, z).$$

Experiments

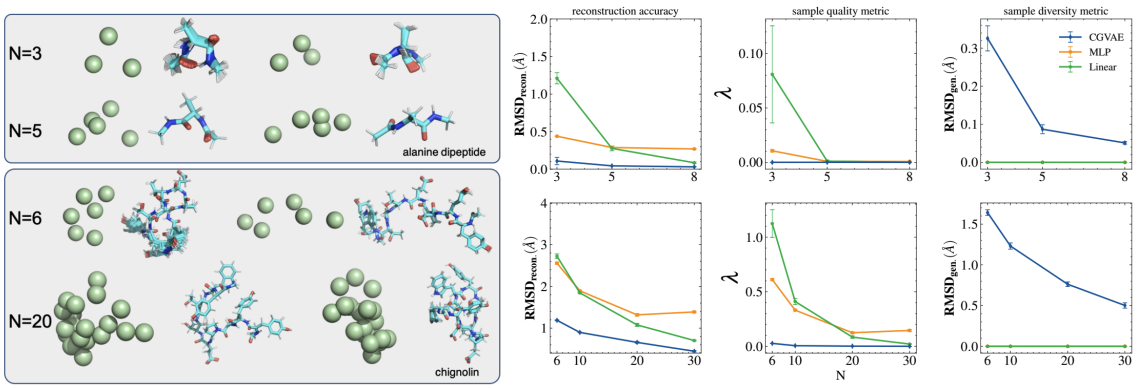


Figure 5: **Left**: Examples of CG and generated FG geometries of alanine dipeptide (top) and chignolin (bottom) generated by CGVAE. The geometry visualizations are created by PyMol (DeLano). **Right**: Benchmarks of reconstructed and generated geometries for heavy-atom structures with different resolution for alanine dipeptide (top) and chignolin (bottom).

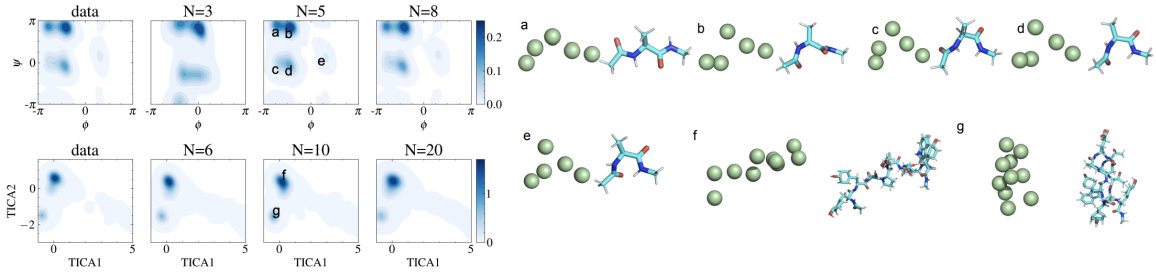


Figure 6: **Left:** 2D Ramachandran plots (top) and TICA plots (bottom) for conformations of alanine dipeptide and chignolin respectively. The structures for the plots are references in the dataset (first column) and samples generated by CGVAE trained with different N (the rest three columns). Figures show that the generated samples recover the important meta-stable states and the distributions agree well with the ground truth. It also shows that the higher resolution model shows better resemblance with the ground truth data. **Right:** Visualization of representative samples in the 2D plots.

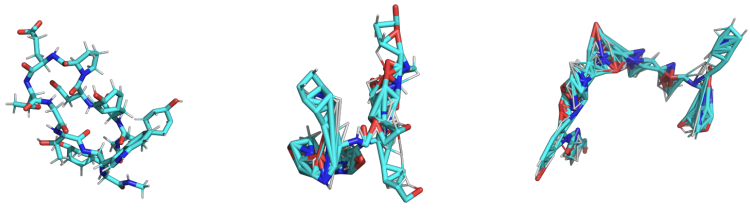


Figure 10: Comparison between generated chignolin samples ($N = 6$) from CGVAE and backmapped samples from baseline methods.