中国人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

CoarsenConf: Equivariant Coarsening with Aggregated Attention for Molecular Conformer Generation

(ICLR 2024 Under Review)

Fanmeng Wang

2023-12-21

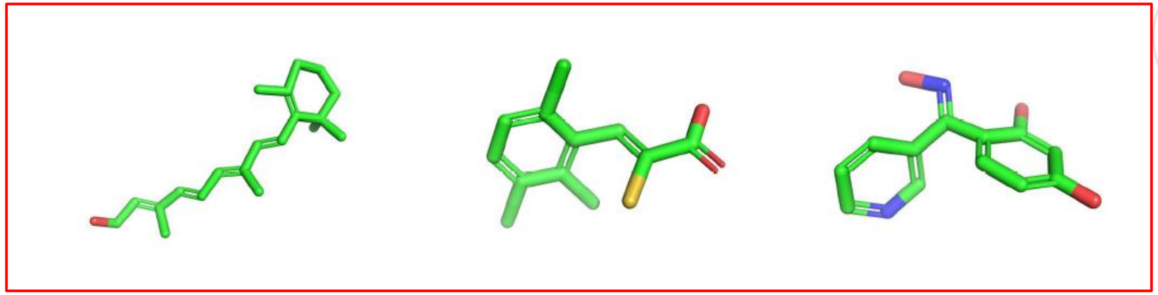
Outline

- Introduction
- Background
- Methods
- Experiment
- Conclusion

- Introduction
- Background
- Methods
- Experiment
- Conclusion

Introduction

- **Molecular conformer generation (MCG)** is a fundamental task in computational chemistry.
  - **Conformers** refer to the stable low-energy 3D molecular structures



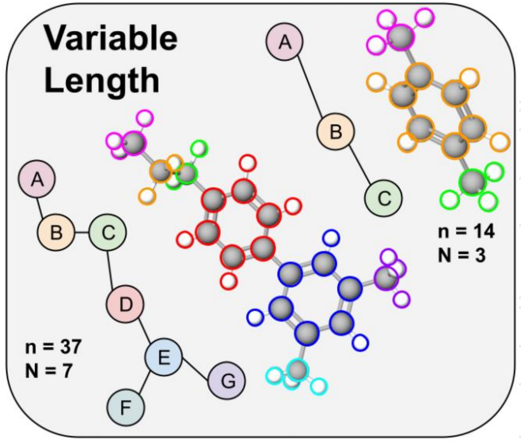
  - **Accurate molecular conformations** are important for various applications that depend on precise spatial and geometric qualities, including drug discovery and protein docking.

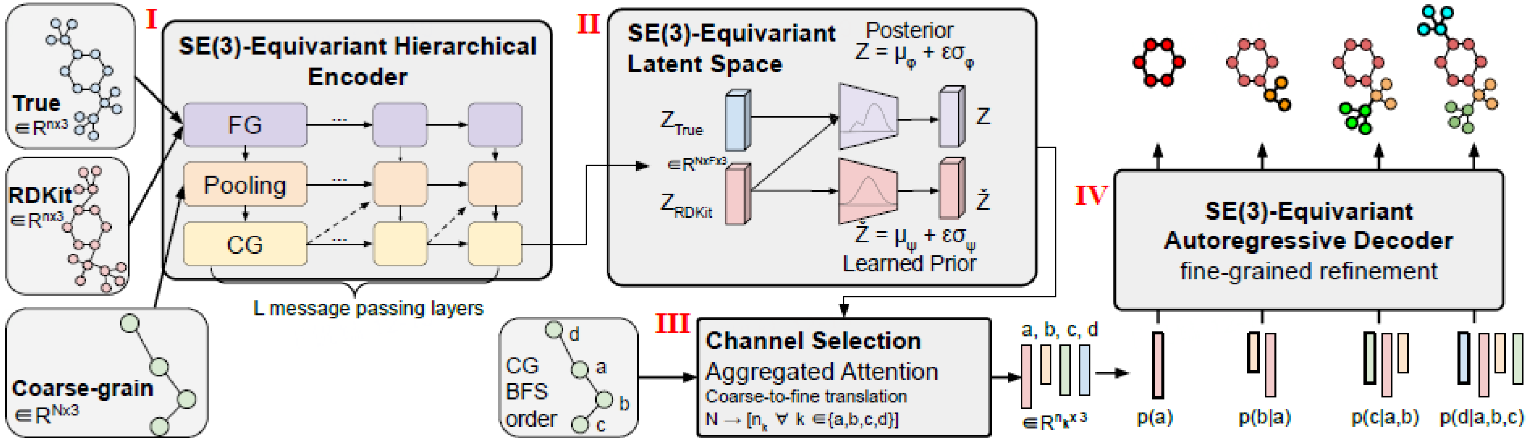
Introduction

- **Exciting Methods** for Molecular Conformation Generation
  - **Traditional physics-based methods**
    - a trade-off between speed and accuracy.
  - **Quantum mechanical methods**
    - more accurate but computationally slow.
  - **Stochastic cheminformatics-based methods** like RDKit ETKDG
    - more efficient but less accurate results.
  - **Existing generative MCG ML-based models**
    - fail to fully leverage the full geometric information inherent to the problem.

Introduction

- **CoarsenConf: an SE(3)-equivariant hierarchical VAE**
  - **Variable-length coarse-to-fine generation strategy**: coarse-grains molecules based on torsional angles, and create a flexible subgraph-level representation from corresponding fine-grained atom coordinates by an Aggregated Attention strategy.
  - CoarsenConf learns a **coarsegrained / subgraph-level latent distribution** for SE(3)-equivariant conformer generation.



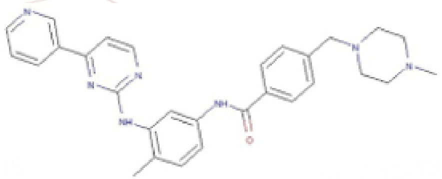


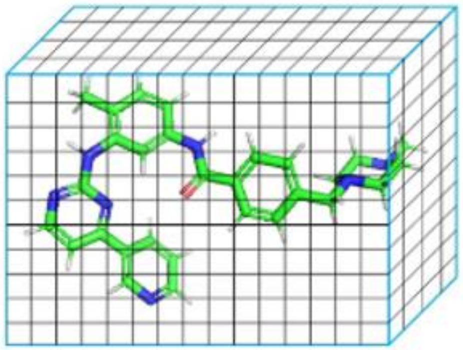
- ➢ Introduction
- ➢ **Background**
- ➢ Methods
- ➢ Experiment
- ➢ Conclusion

Background

Notations

- Each molecule is represented by a **graph** $G = (V, E)$ where V is the set of vertices representing atoms and E is the set of edges representing inter-atomic bonds.
- In addition, each molecular graph is expanded to incorporate **auxiliary edges** connecting all atoms within a 4Å radius to enhance long-range interactions in message passing.
- The **full molecular conformation** is represented by the coordinate matrix $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}|\times 3}$



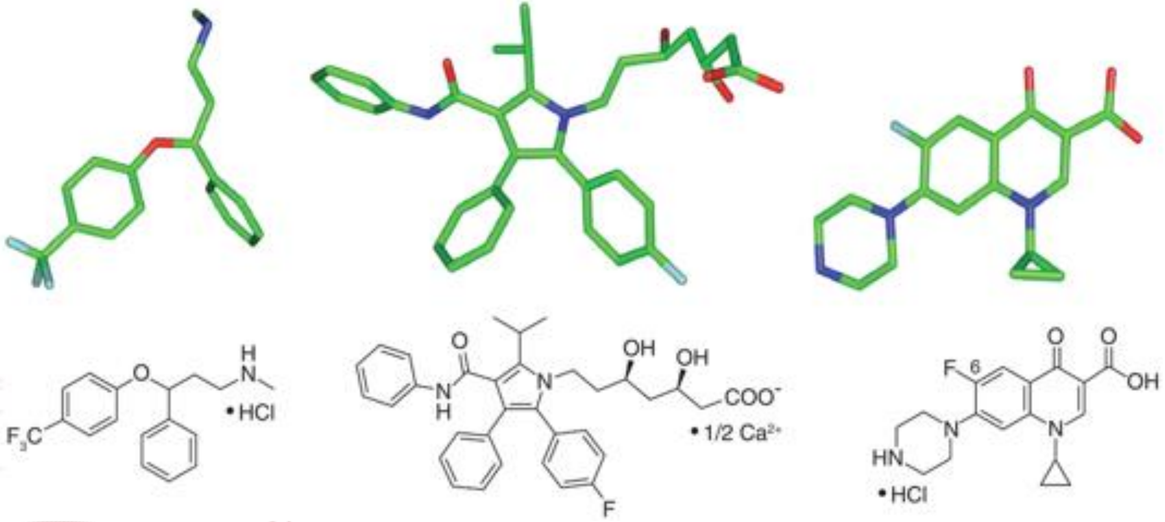


Molecular Graph   Moelcular Conformation

Background

- **Problem Definition**

- **Molecular conformation generation** is a conditional generative process that aims to model the conditional distribution of 3D molecular conformations ***X***, given the 2D molecule graph ***G***, i.e., $p(\boldsymbol{X}|G)$



Background

■ Problem Definition

- ➢ Previous works have showed that RDKit is highly effective at generating conformations with **correct bond distances** and, as a result, can constrain the problem to a diffusion process over only torsion angles.
- ➢ In this paper, we thus formalize MCG as modeling the conditional distribution $p(\boldsymbol{X}|\mathcal{R})$ where $\mathcal{R}$ is the **RDKit generated atomic coordinates**, as we use RDKit as a building block to provide **an approximation starting** from only 2D information.

$$p(\boldsymbol{X}|G) \Longrightarrow p(\boldsymbol{X}|\mathcal{R})$$

$$\mathcal{R} = \text{RDKit ETKDG}(G)$$

Background

■ Molecular Coarse-graining

- ➢ Molecular coarse-graining refers to the simplification of a molecule representation by grouping the **fine-grained (FG) atoms** in the original structure into **individual coarse-grained (CG) beads** with a rule-based mapping.

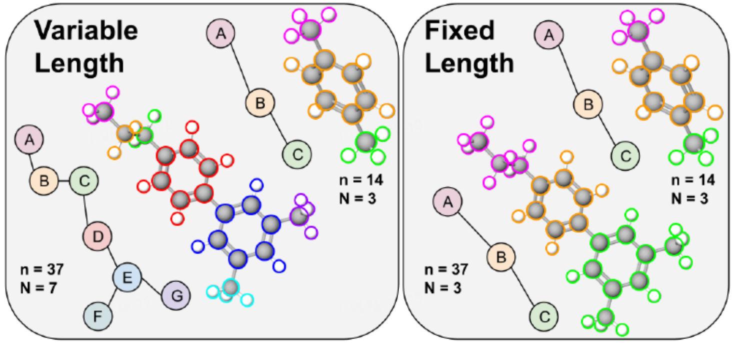


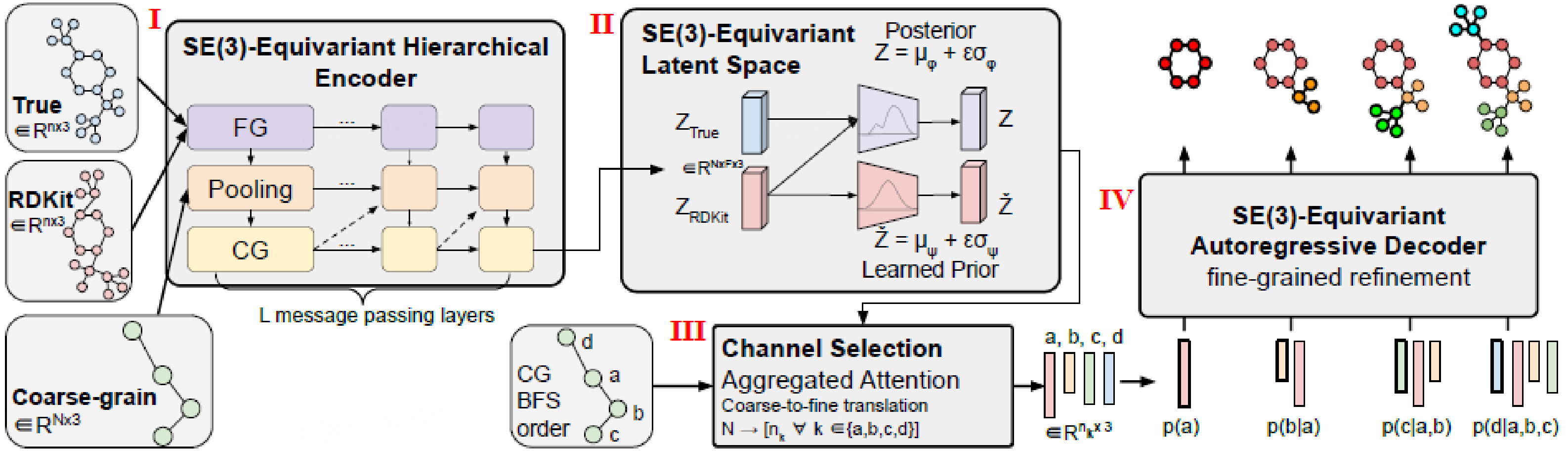
Figure 1: **Learning flexible coarse-grained representations.** CoarsenConf is the first model to employ variable-length (on left) coarse-graining. Each input molecule (n fine-grained (FG) atoms) can be represented by a different number of coarse-grained (CG) nodes N, thus accommodating diverse molecular sizes. In contrast, prior approaches rely on fixed-length (on right) coarse-graining, thereby forcing all molecules to possess the same number of CG nodes. Variable-length coarse-graining enhances the model's ability to create better learned representations across molecules of different sizes and geometries. The molecules on the left are coarsened along torsional angles.

- ➢ In this work, we coarse-grains the given molecule based on torsional angles, thus creating a **flexible variable-length CG representation**.

- Introduction
- Background
- Methods
- Experiment
- Conclusion

Method

- CoarsenConf is a **conditional generative model** that learns $p(X|R)$ where X is the **low-energy 3D conformation**, and R is the **RDKit approximate conformation**.

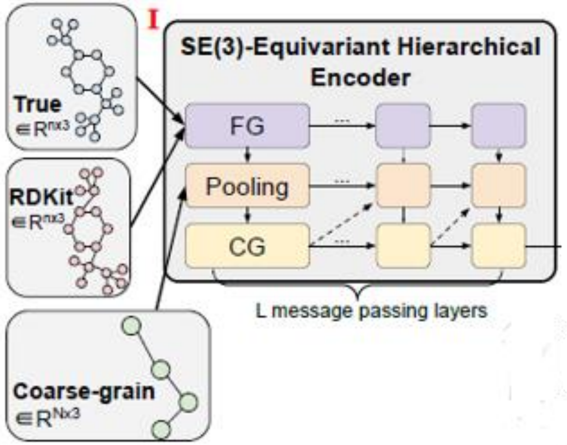


An SE(3)-equivariant hierarchical VAE architecture

Method: 3D Infomax

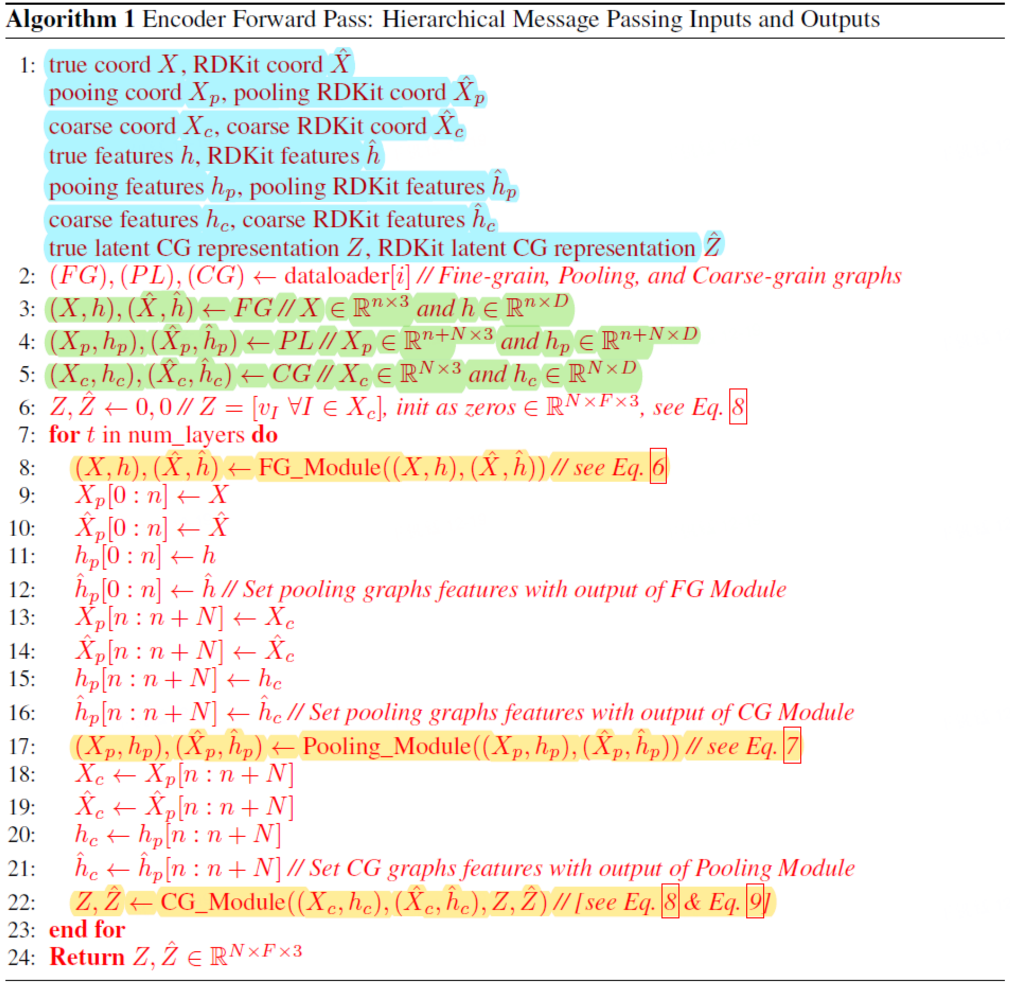
■ Encoder Architecture

- ➢ The encoder module takes in **SE(3)-equivariant atomistic coordinates** $x \in R^{n \times 3}$ and **SE(3)-invariant atom features** $h \in R^{n \times D}$ from the ground truth and RDKit conformer and creates coarse-grained latent representations for each Z and $\tilde{Z} \in R^{N \times F \times 3}$ where N is the number of CG beads, and F is the latent dimensions.
- ➢ A single encoder layer is composed of three modules: fine-grained, pooling, and coarse-grained.



Method: 3D Infomax

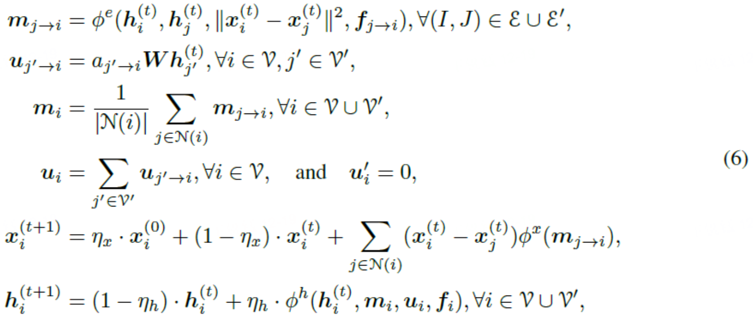
■ Encoder Architecture



Algorithm 1 Encoder Forward Pass: Hierarchical Message Passing Inputs and Outputs

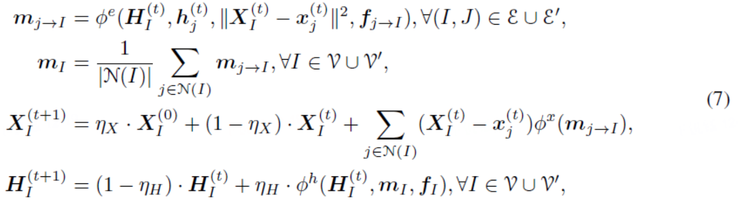
```
true coord $X$, RDKit coord $\hat{X}$
pooing coord $X_p$, pooling RDKit coord $\hat{X}_p$
coarse coord $X_c$, coarse RDKit coord $\hat{X}_c$
true features $h$, RDKit features $\hat{h}$
pooing features $h_p$, pooling RDKit features $\hat{h}_p$
coarse features $h_c$, coarse RDKit features $\hat{h}_c$
true latent CG representation $Z$, RDKit latent CG representation $\hat{Z}$
$(FG), (PL), (CG) \leftarrow$ dataloader$[i]$ // Fine-grain, Pooling, and Coarse-grain graphs
$(X, h), (\hat{X}, \hat{h}) \leftarrow FG$ // $X \in \mathbb{R}^{n\times 3}$ and $h \in \mathbb{R}^{n\times D}$
$(X_p, h_p), (\hat{X}_p, \hat{h}_p) \leftarrow PL$ // $X_p \in \mathbb{R}^{n+N\times 3}$ and $h_p \in \mathbb{R}^{n+N\times D}$
$(X_c, h_c), (\hat{X}_c, \hat{h}_c) \leftarrow CG$ // $X_c \in \mathbb{R}^{N\times 3}$ and $h_c \in \mathbb{R}^{N\times D}$
$Z, \hat{Z} \leftarrow 0, 0$ // $Z = [v_I \; \forall I \in X_c]$, init as zeros $\in \mathbb{R}^{N\times F\times 3}$, see Eq. 8
for $t$ in num_layers do
    $(X, h), (\hat{X}, \hat{h}) \leftarrow$ FG_Module$((X, h), (\hat{X}, \hat{h}))$ // see Eq. 6
    $X_p[0:n] \leftarrow X$
    $\hat{X}_p[0:n] \leftarrow \hat{X}$
    $h_p[0:n] \leftarrow h$
    $\hat{h}_p[0:n] \leftarrow \hat{h}$ // Set pooling graphs features with output of FG Module
    $X_p[n:n+N] \leftarrow X_c$
    $\hat{X}_p[n:n+N] \leftarrow \hat{X}_c$
    $h_p[n:n+N] \leftarrow h_c$
    $\hat{h}_p[n:n+N] \leftarrow \hat{h}_c$ // Set pooling graphs features with output of CG Module
    $(X_p, h_p), (\hat{X}_p, \hat{h}_p) \leftarrow$ Pooling_Module$((X_p, h_p), (\hat{X}_p, \hat{h}_p))$ // see Eq. 7
    $X_c \leftarrow X_p[n:n+N]$
    $\hat{X}_c \leftarrow \hat{X}_p[n:n+N]$
    $h_c \leftarrow h_p[n:n+N]$
    $\hat{h}_c \leftarrow \hat{h}_p[n:n+N]$ // Set CG graphs features with output of Pooling Module
    $Z, \hat{Z} \leftarrow$ CG_Module$((X_c, h_c), (\hat{X}_c, \hat{h}_c), Z, \hat{Z})$ // [see Eq. 8 & Eq. 9]
end for
Return $Z, \hat{Z} \in \mathbb{R}^{N\times F\times 3}$
```

FG Module



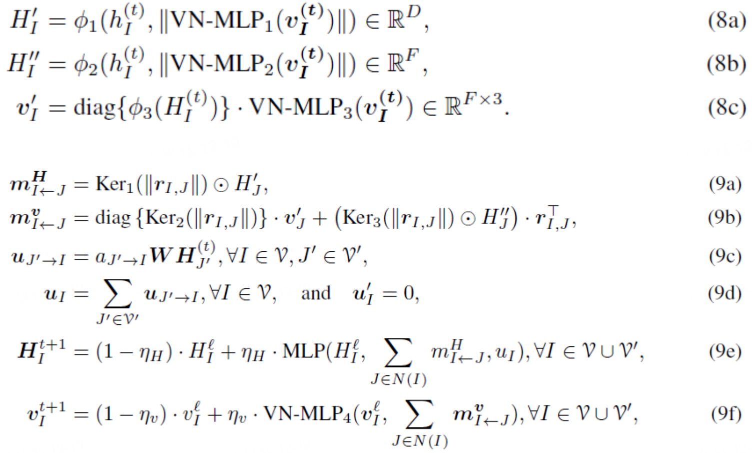
$$
\begin{aligned}
\boldsymbol{m}_{j\to i} &= \phi^e(\boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)}, \|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\|^2, \boldsymbol{f}_{j\to i}), \forall (I, J) \in \mathcal{E} \cup \mathcal{E}', \\
\boldsymbol{u}_{j'\to i} &= a_{j'\to i}\boldsymbol{W}\boldsymbol{h}_{j'}^{(t)}, \forall i \in \mathcal{V}, j' \in \mathcal{V}', \\
\boldsymbol{m}_i &= \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \boldsymbol{m}_{j\to i}, \forall i \in \mathcal{V}\cup\mathcal{V}', \\
\boldsymbol{u}_i &= \sum_{j'\in\mathcal{V}'} \boldsymbol{u}_{j'\to i}, \forall i \in \mathcal{V}, \quad \text{and} \quad \boldsymbol{u}_i' = 0, \qquad (6) \\
\boldsymbol{x}_i^{(t+1)} &= \eta_x \cdot \boldsymbol{x}_i^{(0)} + (1-\eta_x)\cdot \boldsymbol{x}_i^{(t)} + \sum_{j\in\mathcal{N}(i)} (\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)})\phi^x(\boldsymbol{m}_{j\to i}), \\
\boldsymbol{h}_i^{(t+1)} &= (1-\eta_h)\cdot \boldsymbol{h}_i^{(t)} + \eta_h \cdot \phi^h(\boldsymbol{h}_i^{(t)}, \boldsymbol{m}_i, \boldsymbol{u}_i, \boldsymbol{f}_i), \forall i \in \mathcal{V}\cup\mathcal{V}',
\end{aligned}
$$

Pooling Module



$$
\begin{aligned}
\boldsymbol{m}_{j\to I} &= \phi^e(\boldsymbol{H}_I^{(t)}, \boldsymbol{h}_j^{(t)}, \|\boldsymbol{X}_I^{(t)} - \boldsymbol{x}_j^{(t)}\|^2, \boldsymbol{f}_{j\to I}), \forall (I, J) \in \mathcal{E} \cup \mathcal{E}', \\
\boldsymbol{m}_I &= \frac{1}{|\mathcal{N}(I)|}\sum_{j\in\mathcal{N}(I)} \boldsymbol{m}_{j\to I}, \forall I \in \mathcal{V}\cup\mathcal{V}', \qquad (7) \\
\boldsymbol{X}_I^{(t+1)} &= \eta_X \cdot \boldsymbol{X}_I^{(0)} + (1-\eta_X)\cdot \boldsymbol{X}_I^{(t)} + \sum_{j\in\mathcal{N}(I)} (\boldsymbol{X}_I^{(t)} - \boldsymbol{x}_j^{(t)})\phi^x(\boldsymbol{m}_{j\to I}), \\
\boldsymbol{H}_I^{(t+1)} &= (1-\eta_H)\cdot \boldsymbol{H}_I^{(t)} + \eta_H \cdot \phi^h(\boldsymbol{H}_I^{(t)}, \boldsymbol{m}_I, \boldsymbol{f}_I), \forall I \in \mathcal{V}\cup\mathcal{V}',
\end{aligned}
$$

CG Module



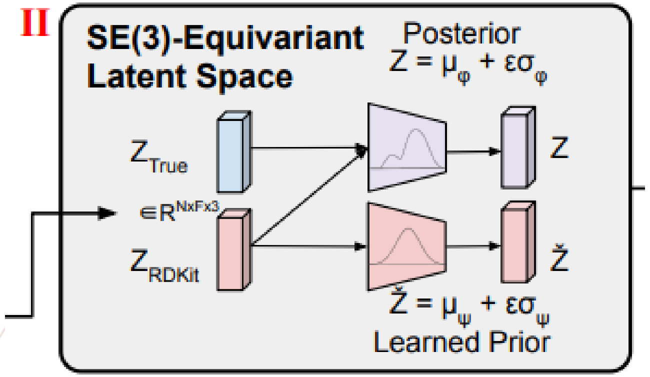
$$
\begin{aligned}
H_I' &= \phi_1(h_I^{(t)}, \|\text{VN-MLP}_1(\boldsymbol{v}_I^{(t)})\|) \in \mathbb{R}^D, && \text{(8a)} \\
H_I'' &= \phi_2(h_I^{(t)}, \|\text{VN-MLP}_2(\boldsymbol{v}_I^{(t)})\|) \in \mathbb{R}^F, && \text{(8b)} \\
\boldsymbol{v}_I' &= \text{diag}\{\phi_3(H_I^{(t)})\} \cdot \text{VN-MLP}_3(\boldsymbol{v}_I^{(t)}) \in \mathbb{R}^{F\times 3}. && \text{(8c)}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{m}_{I\leftarrow J}^{\boldsymbol{H}} &= \text{Ker}_1(\|\boldsymbol{r}_{I,J}\|) \odot H_J', && \text{(9a)} \\
\boldsymbol{m}_{I\leftarrow J}^{\boldsymbol{v}} &= \text{diag}\left\{\text{Ker}_2(\|\boldsymbol{r}_{I,J}\|)\right\} \cdot \boldsymbol{v}_J' + \left(\text{Ker}_3(\|\boldsymbol{r}_{I,J}\|) \odot H_J''\right) \cdot \boldsymbol{r}_{I,J}^\top, && \text{(9b)} \\
\boldsymbol{u}_{J'\to I} &= a_{J'\to I}\boldsymbol{W}\boldsymbol{H}_{J'}^{(t)}, \forall I \in \mathcal{V}, J' \in \mathcal{V}', && \text{(9c)} \\
\boldsymbol{u}_I &= \sum_{J'\in\mathcal{V}'} \boldsymbol{u}_{J'\to I}, \forall I \in \mathcal{V}, \quad \text{and} \quad \boldsymbol{u}_I' = 0, && \text{(9d)} \\
\boldsymbol{H}_I^{t+1} &= (1-\eta_H)\cdot H_I^\ell + \eta_H \cdot \text{MLP}(H_I^\ell, \sum_{J\in N(I)} m_{I\leftarrow J}^H, u_I), \forall I \in \mathcal{V}\cup\mathcal{V}', && \text{(9e)} \\
\boldsymbol{v}_I^{t+1} &= (1-\eta_v)\cdot \boldsymbol{v}_I^\ell + \eta_v \cdot \text{VN-MLP}_4(\boldsymbol{v}_I^\ell, \sum_{J\in N(I)} \boldsymbol{m}_{I\leftarrow J}^{\boldsymbol{v}}), \forall I \in \mathcal{V}\cup\mathcal{V}', && \text{(9f)}
\end{aligned}
$$

Method: 3D Infomax

■ Equivariant Latent Space

➢ The **conditional posterior** is parameterized with both the ground truth and RDKit approximation, whereas the **learned conditional** prior only uses the RDKit.



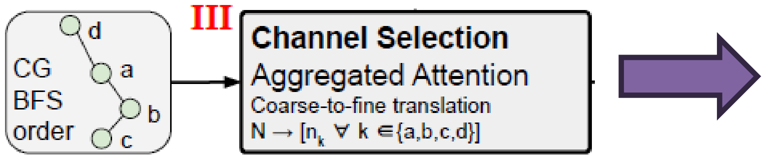
$$\begin{aligned} \text{Posterior}: \boldsymbol{\mu}_{\phi} &= \text{VN-MLP}(Z, \tilde{Z}), \quad \log(\boldsymbol{\sigma}_{\phi}^{2}) = \text{MLP}(Z, \tilde{Z}), \\ \text{Prior}: \boldsymbol{\mu}_{\psi} &= \text{VN-MLP}(\tilde{Z}), \quad \log(\boldsymbol{\sigma}_{\psi}^{2}) = \text{MLP}(\tilde{Z}). \end{aligned} \tag{2}$$

Method: 3D Infomax

■ Decoder Architecture: Channel Selection

➢ Given latent representation Z is still in CG space, we need to perform variable-length backmapping to convert back to FG space so that we can further refine the atom coordinates to generate the low energy conformer

$$X \leftarrow \text{Aggregated\_Attention}(\text{KV} = Z, \text{Q} = \hat{X})$$



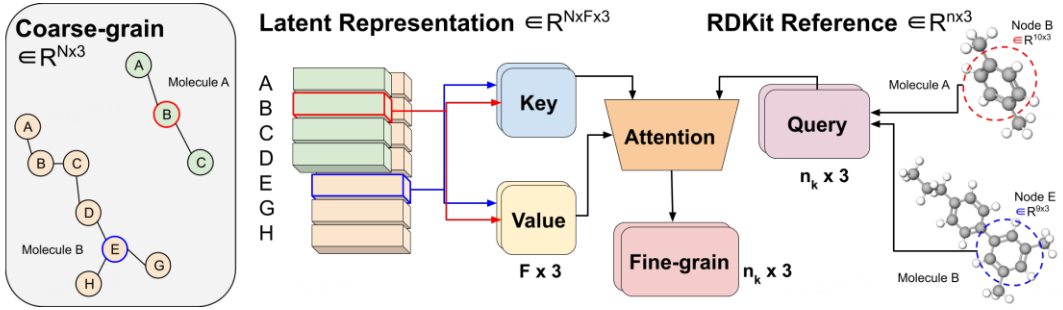
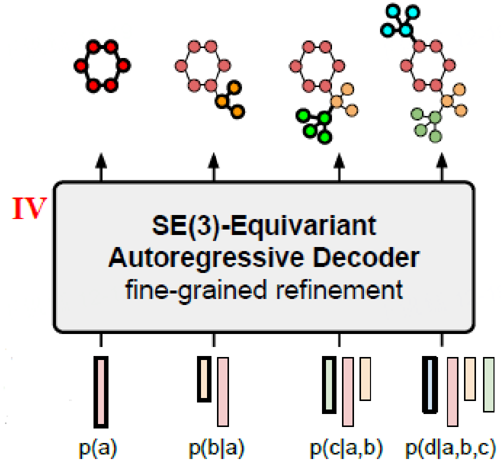


Figure 3: **Variable-length coarse-to-fine backmapping via Aggregated Attention.** The highlighted latent beads of two independent molecules are attended to by the respective fine-grained queries in a batched manner (see red and blue), to generate FG coordinates in the desired shape (matching input queries on right). The single-head attention operation uses the latent vectors of each CG bead $Z_k \in R^{F\times3}$ for each molecule as the keys and values, with an embedding dimension of 3 to match the x, y, z coordinates. The query vectors are the FG subset of the respective RDKit conformers, corresponding to each CG bead $\in R^{n_k\times3}$. We know a priori how many FG atoms correspond to a certain CG bead (n_k). Aggregated Attention learns the optimal blending of CG features for FG reconstruction by aggregating 3D segments of FG information to form our latent query.

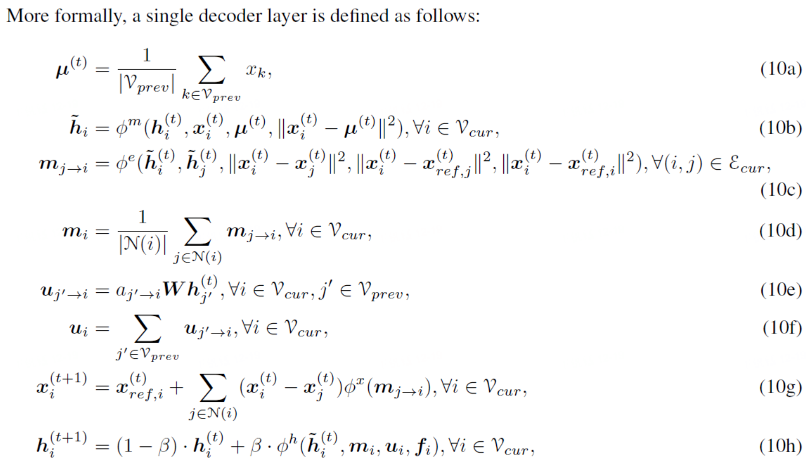
Method: 3D Infomax

■ Decoder Architecture: Channel Selection

- ➢ The decoder architecture is similar to the EGNN-based FG module in the encoder, but we just learn to predict the **difference** between the RDKit conformations and ground truth conformations.
- ➢ We simplify the learning objective by setting X = R + ΔX, and learn the optimal distortion ΔX from the RDKit approximation.



SE(3)-Equivariant Autoregressive Decoder
fine-grained refinement



More formally, a single decoder layer is defined as follows:

$$\boldsymbol{\mu}^{(t)} = \frac{1}{|\mathcal{V}_{prev}|} \sum_{k \in \mathcal{V}_{prev}} x_k, \tag{10a}$$

$$\tilde{\boldsymbol{h}}_i = \phi^m(\boldsymbol{h}_i^{(t)}, \boldsymbol{x}_i^{(t)}, \boldsymbol{\mu}^{(t)}, \|\boldsymbol{x}_i^{(t)} - \boldsymbol{\mu}^{(t)}\|^2), \forall i \in \mathcal{V}_{cur}, \tag{10b}$$

$$\boldsymbol{m}_{j \to i} = \phi^e(\tilde{\boldsymbol{h}}_i^{(t)}, \tilde{\boldsymbol{h}}_j^{(t)}, \|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\|^2, \|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_{ref,j}^{(t)}\|^2, \|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_{ref,i}^{(t)}\|^2), \forall (i,j) \in \mathcal{E}_{cur}, \tag{10c}$$

$$\boldsymbol{m}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{j \to i}, \forall i \in \mathcal{V}_{cur}, \tag{10d}$$

$$\boldsymbol{u}_{j' \to i} = a_{j' \to i} \boldsymbol{W} \boldsymbol{h}_{j'}^{(t)}, \forall i \in \mathcal{V}_{cur}, j' \in \mathcal{V}_{prev}, \tag{10e}$$

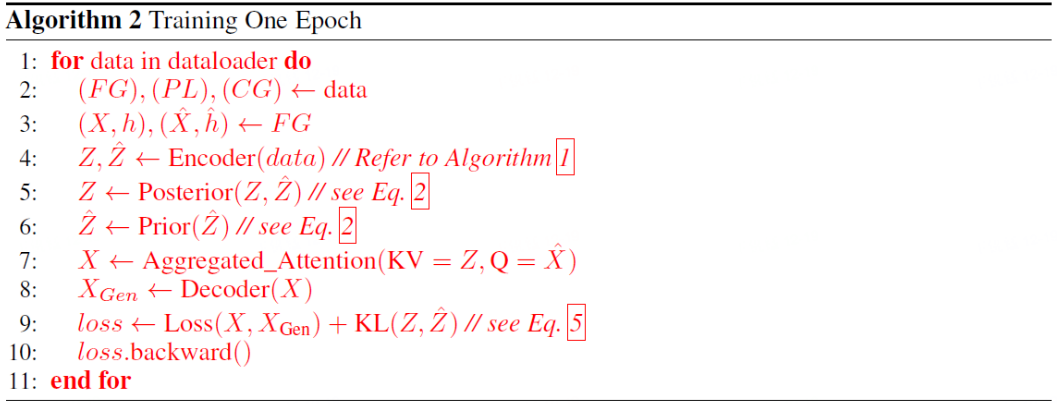
$$\boldsymbol{u}_i = \sum_{j' \in \mathcal{V}_{prev}} \boldsymbol{u}_{j' \to i}, \forall i \in \mathcal{V}_{cur}, \tag{10f}$$

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_{ref,i}^{(t)} + \sum_{j \in \mathcal{N}(i)} (\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}) \phi^x(\boldsymbol{m}_{j \to i}), \forall i \in \mathcal{V}_{cur}, \tag{10g}$$

$$\boldsymbol{h}_i^{(t+1)} = (1 - \beta) \cdot \boldsymbol{h}_i^{(t)} + \beta \cdot \phi^h(\tilde{\boldsymbol{h}}_i^{(t)}, \boldsymbol{m}_i, \boldsymbol{u}_i, \boldsymbol{f}_i), \forall i \in \mathcal{V}_{cur}, \tag{10h}$$

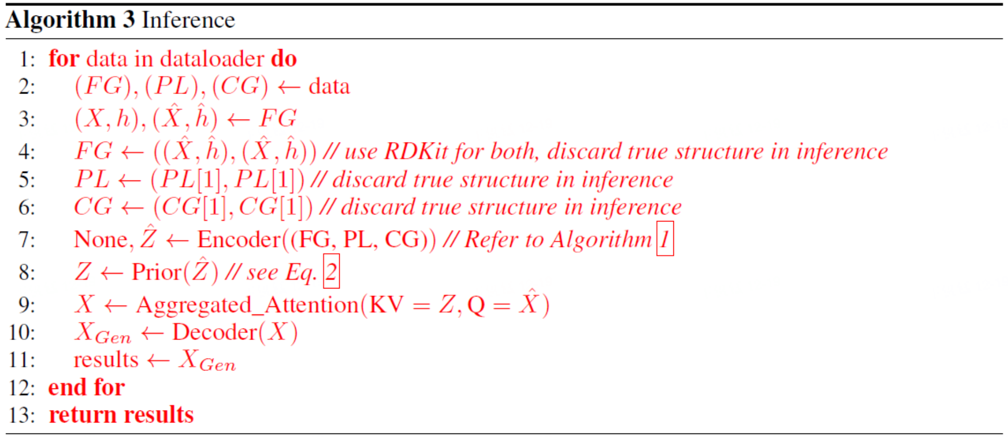
Method: 3D Infomax

General algorithms



Algorithm 2 Training One Epoch

```
for data in dataloader do
    $(FG), (PL), (CG) \leftarrow$ data
    $(X, h), (\hat{X}, \hat{h}) \leftarrow FG$
    $Z, \hat{Z} \leftarrow \text{Encoder}(data)$ // Refer to Algorithm 1
    $Z \leftarrow \text{Posterior}(Z, \hat{Z})$ // see Eq. 2
    $\hat{Z} \leftarrow \text{Prior}(\hat{Z})$ // see Eq. 2
    $X \leftarrow \text{Aggregated\_Attention}(\text{KV} = Z, \text{Q} = \hat{X})$
    $X_{Gen} \leftarrow \text{Decoder}(X)$
    $loss \leftarrow \text{Loss}(X, X_{\text{Gen}}) + \text{KL}(Z, \hat{Z})$ // see Eq. 5
    $loss.\text{backward}()$
end for
```



Algorithm 3 Inference

```
for data in dataloader do
    $(FG), (PL), (CG) \leftarrow$ data
    $(X, h), (\hat{X}, \hat{h}) \leftarrow FG$
    $FG \leftarrow ((\hat{X}, \hat{h}), (\hat{X}, \hat{h}))$ // use RDKit for both, discard true structure in inference
    $PL \leftarrow (PL[1], PL[1])$ // discard true structure in inference
    $CG \leftarrow (CG[1], CG[1])$ // discard true structure in inference
    None, $\hat{Z} \leftarrow$ Encoder((FG, PL, CG)) // Refer to Algorithm 1
    $Z \leftarrow \text{Prior}(\hat{Z})$ // see Eq. 2
    $X \leftarrow \text{Aggregated\_Attention}(\text{KV} = Z, \text{Q} = \hat{X})$
    $X_{Gen} \leftarrow \text{Decoder}(X)$
    results $\leftarrow X_{Gen}$
end for
return results
```

- Loss function (one-to-one loss)

$$\boxed{\text{MSE}(\mathcal{A}(X, X_{true})) + \beta_1 D_{KL}(q_\phi(z|X, \mathcal{R}) \parallel p_\psi(z|\mathcal{R}))} + \beta_2 \frac{1}{|\mathcal{E}^*|} \sum_{(i,j)\in\mathcal{E}^*} ||r_{ij} - r_{ij}^{true}||^2, \qquad (5)$$

- Optimal Transport reduces compute requirements

$$\mathcal{L}_{OT} = \min_{\mathrm{T}\in\mathcal{Q}_{K,L}} \sum_{k,l} T_{kl}\mathcal{L}(\mathcal{C}_k, \mathcal{C}_l^*),$$

$$\mathcal{L}(\mathcal{C}_k, \mathcal{C}_l^*) = \text{MSE}(\mathcal{C}_k, \mathcal{C}_l^*) + \text{distance error}(\mathcal{C}_k, \mathcal{C}_l^*),$$

- Introduction
- Background
- Methods
- Experiment
- Conclusion

Experiment

■ GEOM BENCHMARKS: 3D COORDINATE RMSD

- **GEOM dataset:** consisting of QM9 (average 11 atoms, 15 conformers per molecule) and DRUGS (average 44 atoms, 104 per molecule)

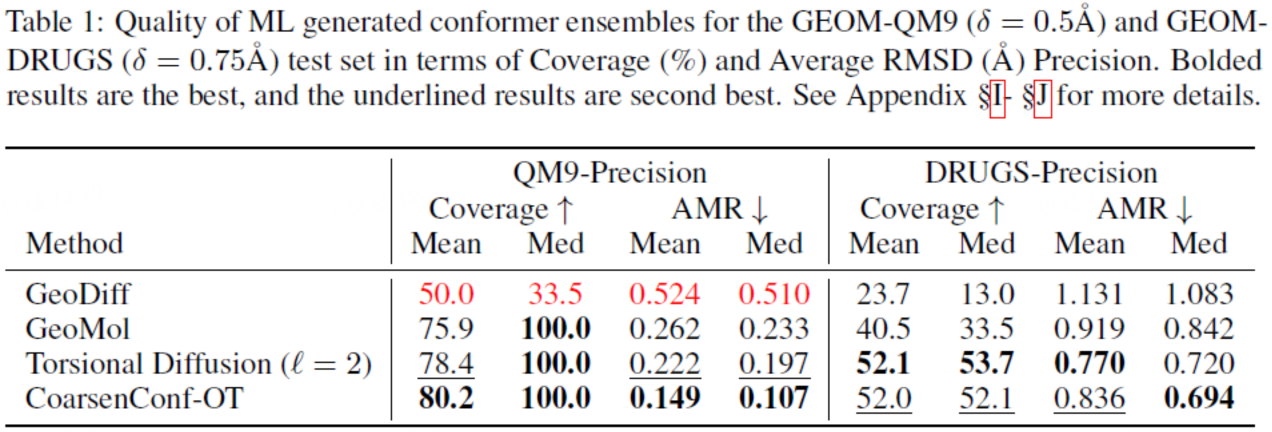
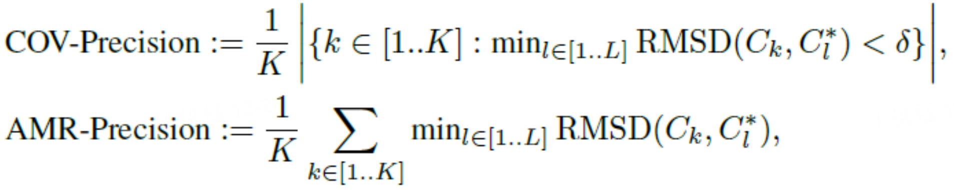


Table 1: Quality of ML generated conformer ensembles for the GEOM-QM9 ($\delta = 0.5$Å) and GEOM-DRUGS ($\delta = 0.75$Å) test set in terms of Coverage (%) and Average RMSD (Å) Precision. Bolded results are the best, and the underlined results are second best. See Appendix §I- §J for more details.

	QM9-Precision				DRUGS-Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
GeoDiff	50.0	33.5	0.524	0.510	23.7	13.0	1.131	1.083
GeoMol	75.9	**100.0**	0.262	0.233	40.5	33.5	0.919	0.842
Torsional Diffusion ($\ell = 2$)	78.4	**100.0**	0.222	0.197	**52.1**	**53.7**	**0.770**	0.720
CoarsenConf-OT	**80.2**	**100.0**	**0.149**	**0.107**	52.0	52.1	0.836	**0.694**

- Metrics:



$$\text{COV-Precision} := \frac{1}{K}\left|\{k \in [1..K] : \min_{l \in [1..L]} \text{RMSD}(C_k, C_l^*) < \delta\}\right|,$$

$$\text{AMR-Precision} := \frac{1}{K}\sum_{k \in [1..K]} \min_{l \in [1..L]} \text{RMSD}(C_k, C_l^*),$$

- We outperform all models on QM9, and yield competitive results on DRUGS when using an optimal transport (OT) loss.

Experiment

■ GEOM BENCHMARKS: PROPERTY PREDICTION

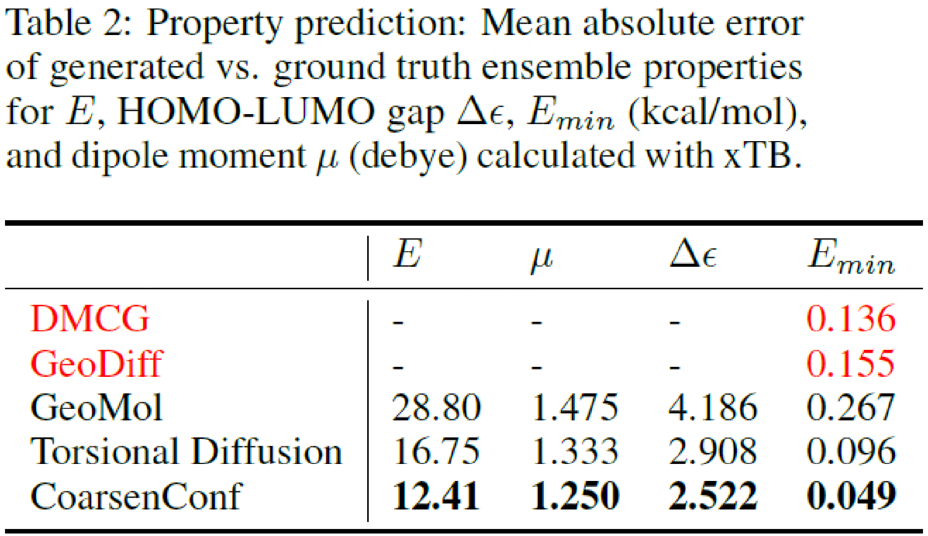


Table 2: Property prediction: Mean absolute error of generated vs. ground truth ensemble properties for E, HOMO-LUMO gap $\Delta\epsilon$, E_{min} (kcal/mol), and dipole moment μ (debye) calculated with xTB.

	E	μ	$\Delta\epsilon$	E_{min}
DMCG	-	-	-	0.136
GeoDiff	-	-	-	0.155
GeoMol	28.80	1.475	4.186	0.267
Torsional Diffusion	16.75	1.333	2.908	0.096
CoarsenConf	**12.41**	**1.250**	**2.522**	**0.049**

➢ CoarsenConf is able to generate **the lowest energy structures** with **the most accurate chemical properties**.

Experiment

Via Autodock Vina's flexible docking simulation

■ FLEXIBLE ORACLE-BASED PROTEIN DOCKING

- ➢ We evaluate MCG models, pretrained on GEOM-DRUGS, using nine protein docking oracle functions provided by the Therapeutics Data Commons (TDC)
- ➢ For each evaluated MCG method, we generate 50 conformers for each of the nine ligands and report the best (lowest) binding affinity.

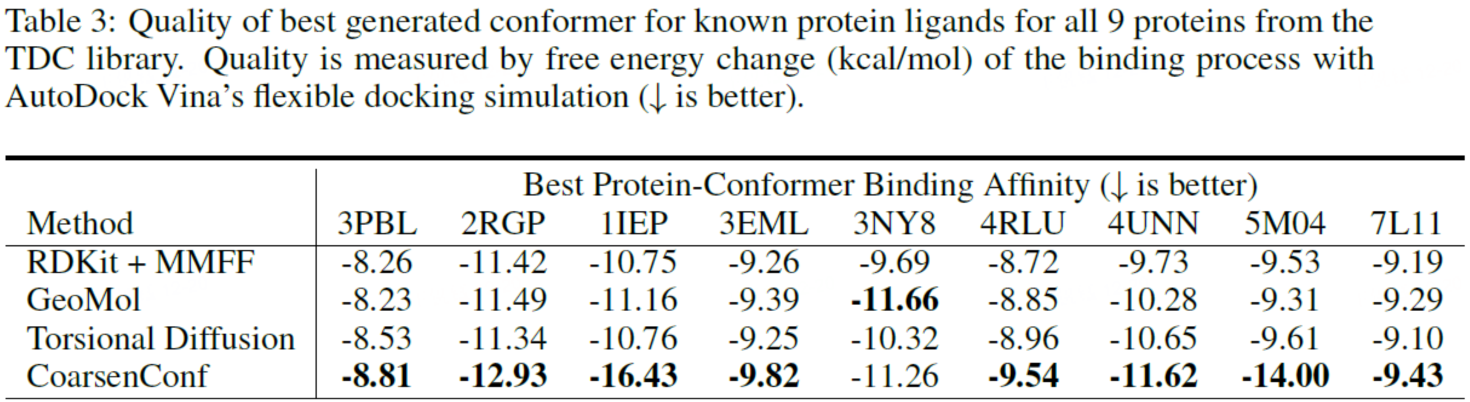


Table 3: Quality of best generated conformer for known protein ligands for all 9 proteins from the TDC library. Quality is measured by free energy change (kcal/mol) of the binding process with AutoDock Vina's flexible docking simulation (↓ is better).

	Best Protein-Conformer Binding Affinity (↓ is better)								
Method	3PBL	2RGP	1IEP	3EML	3NY8	4RLU	4UNN	5M04	7L11
RDKit + MMFF	-8.26	-11.42	-10.75	-9.26	-9.69	-8.72	-9.73	-9.53	-9.19
GeoMol	-8.23	-11.49	-11.16	-9.39	**-11.66**	-8.85	-10.28	-9.31	-9.29
Torsional Diffusion	-8.53	-11.34	-10.76	-9.25	-10.32	-8.96	-10.65	-9.61	-9.10
CoarsenConf	**-8.81**	**-12.93**	**-16.43**	**-9.82**	-11.26	**-9.54**	**-11.62**	**-14.00**	**-9.43**

- ➢ CoarsenConf significantly outperforms prior MCG methods on the TDC oracle-based affinity prediction task.

- Introduction
- Background
- Methods
- **Experiment**
- Conclusion

Conclusion

- We present CoarsenConf, a novel approach for robust molecular conformer generation that combines an SE(3)-equivariant hierarchical VAE with geometric coarse-graining strategy for accurate conformer generation.
- By utilizing easy-to-obtain approximate conformations generated by RDKit, our model effectively learns to generate low-energy conformers.
- Our experiments demonstrate the effectiveness of CoarsenConf compared to existing methods.

Thank You for listening!

Fanmeng Wang

2023-12-21