GEOMOL: Torsional Geometric Generation of Molecular 3D Conformer Ensembles


* equal contribution and first authorship
Motivation

• Recent breakthrough in **Protein folding**
Motivation

• How about molecules?
Motivation

- Drug - target interactions depend on the 3D structure of both the protein and the ligand

Figure 3: The interaction between SCHEMBL16362922 and the MAP kinase-interacting serine/threonine-protein kinase 2. The protein is shown in yellow and the small molecule is shown in green.

Fig source: Interpretable Drug Target Prediction Using Deep Neural Representation , IJCAI'18
Motivation

• Molecules have potentially thousands of stable conformers.
Motivation

• Molecules have potentially thousands of stable conformers.

• **Goal**: predict (all) low energy conformers of an input molecular graph.
Motivation

• Molecules have potentially thousands of stable conformers.

• **Goal**: predict (all) low energy conformers of an input molecular graph.
  
  • Model rigid and flexible parts of the 3D structure.
Motivation

- **Goal**: predict (all) low energy conformers of an input molecular graph
- Energy -> implicitly defined by a given dataset
Motivation

• **Goal**: predict (all) low energy conformers of an input molecular graph
  • Energy -> implicitly defined by a given dataset

• **Why?**
  • Faster, computationally efficient and more accurate conformation generation using Message Passing Neural Networks
  • Usable in various 3D downstream tasks, e.g.:
    • Protein - ligand binding
    • Molecular docking poses
    • Generating conformers inside 3D enzyme pockets
    • Generating descriptors for 4D QSAR
  • Intermediate representation for various property predictors (e.g. biological activity); pre-training technique
  • Generalize to unseen structures

Fig source: Wikipedia
Traditional Conformer generation

• **Stochastic methods:**
  - ETKDG/RDKit (distance geometry initialization + subsequent optimization)
  - Molecular dynamics
  - Sampling torsions with genetic algorithms

• **Drawbacks:**
  - Difficult to sample diverse and representative conformers
  - Computationally expensive
Traditional Conformer generation

**Stochastic methods:**
- ETKDG/RDKit (distance geometry initialization + subsequent optimization)
- Molecular dynamics
- Sampling torsions with genetic algorithms

**Systematic methods:**
- Exhaustive search over torsion angles
- Using databases of torsion templates (torsion rules)
- Commercial software: OMEGA

**Drawbacks:**
- Difficult to sample diverse and representative conformers
- Computationally expensive

- Computational prohibitive for structures with large number of rotatable bonds
- Poor generalization to unseen structures
Traditional Conformer generation

**Stochastic methods:**
- ETKDG/RDKit (distance geometry initialization + subsequent optimization)
- Molecular dynamics
- Sampling torsions with genetic algorithms

**Systematic methods:**
- Exhaustive search over torsion angles
- Using databases of torsion templates (torsion rules)
- Commercial software: OMEGA

**Drawbacks:**
- Difficult to sample diverse and representative conformers
- Computational expensive

**Fine-tunning with Force Fields (FF) is often needed**
- Crude approximations of the true energy
- Parameters derived from experiments in physics and chemistry or calculations in quantum mechanics - strong assumptions (simplistic formulas)
- Various limitations, e.g. ability to accurately capture subtle, weak interactions in biomolecules.
- Alternative: computationally expensive methods such as metadynamics with approximate quantum mechanics calculations at each geometry

**Drawbacks:**
- Computational prohibitive for structures with large number of rotatable bonds
- Poor generalization to unseen structures
ML Approaches for Conformer generation

- Multi-stage models: generate distance matrix, then predict coordinates, then fine-tune the conformer [2,3]
- Typically need a FF and/or extra energy model
- Not trainable end-to-end -> error accumulation in multi-stage pipelines
- No explicit handling of classic molecular geometry elements: bond angles, torsions angles, chirality, etc.
- Relies on good distance matrix predictions
- Can't explicitly predict chiral structures or cis/trans conformations


Input: molecular graph (+ random noise)
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings

Predict 1-hop 3D coordinates. Result: bond distances and bond angles.
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings

Predict 1-hop 3D coordinates. Result: bond distances and bond angles.

Assemble local neighboring structures. Result: torsion angles and 3-hop distances

Torsion angle

Neural Net

\( h_X \), \( h_Y \), \( h_{\text{mol}} \)

\( \alpha \)
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings

Predict 1-hop 3D coordinates. Result: bond distances and bond angles.

Assemble local neighboring structures. Result: torsion angles and 3-hop distances

Training loss: 1,2,3-hop distance loss; bond angle & torsion angle loss
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings

Predict 1-hop 3D coordinates. Result: bond distances and bond angles.

Assemble local neighboring structures. Result: torsion angles and 3-hop distances

Training loss: 1,2,3-hop distance loss; bond angle & torsion angle loss

Full conformer assembly

Test time (only). Involves: Deterministic ring corrections
GeoMol - Overview

Input: molecular graph (+ random noise)

Message Passing Neural Network

Atom embeddings

Predict 1-hop 3D coordinates. Result: bond distances and bond angles.

Assemble local neighboring structures. Result: torsion angles and 3-hop distances

Training loss: 1,2,3-hop distance loss; bond angle & torsion angle loss

Full conformer assembly

Test time (only). Involves: Deterministic ring corrections

Repeat N times to generate N conformers
**GeoMol - Overview**

Input: molecular graph (+ random noise) 

Message Passing Neural Network 

Atom embeddings 

Predict 1-hop 3D coordinates. Result: bond distances and bond angles. 

Assemble local neighboring structures. Result: torsion angles and 3-hop distances 

Training loss: 1, 2, 3-hop distance loss; bond angle & torsion angle loss 

Full conformer assembly 

Test time (only). Involves: Deterministic ring corrections 

Repeat N times to generate N conformers 

**Contributions:**

- Explicit prediction of bond distances, bond angles and torsion angles. 
- Trainable end-to-end & non-autoregressive: joint prediction of all atom 3D coordinates from the molecular graph 
- 3D coordinates predicted SE(3)-invariantly, i.e. to any global rotation/translation 
- Tetrahedral chiral centers are predicted exactly; no iterative optimization necessary as with traditional distance geometry approaches 
- Diversity of generated conformers: achieved using a tailored Wasserstein generative loss
Local Structure (LS) Prediction

For each non-terminal atom, predict 3D coordinates of all its 1-hop neighbors (atom local 3D environment)

**Result:** bond distances and bond angles.
Local Structure (LS) Prediction

For each non-terminal atom $X$, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming $X$ is placed in 0:

$$f(h_{T_1}, \ldots, h_{T_n}; \hat{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n}$$

Result: bond distances and bond angles.
Local Structure (LS) Prediction

For each non-terminal atom $X$, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming $X$ is placed in $0$: $f(h_{T_1}, \ldots, h_{T_n}; \tilde{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n}$

**Result:** bond distances and bond angles.

**Challenges:**

1) Equivariant prediction with respect to any permutation of neighbors: $f(h_{T_{\pi(1)}}, \ldots, h_{T_{\pi(n)}}; \tilde{h}_X) = (p_{\pi(1)}, \ldots, p_{\pi(n)}), \forall \pi \in S_n$
Local Structure (LS) Prediction

For each non-terminal atom $X$, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming $X$ is placed in $0$:

$$f(h_{T_1}, \ldots, h_{T_n}, \hat{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n}$$

**Result:** bond distances and bond angles.

**Challenges:**

1) Equivariant prediction with respect to any permutation of neighbors: $f(h_{T_{\pi(1)}}, \ldots, h_{T_{\pi(n)}}, \hat{h}_X) = (p_{\pi(1)}, \ldots, p_{\pi(n)}), \forall \pi \in S_n$

2) Bond distances should match,
   - i.e. $XT_1$ should have the same length predicted from the LS of $X$ or from the LS of $T_1$
Local Structure (LS) Prediction

For each non-terminal atom $X$, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming $X$ is placed in $0$:

$$f(h_{T_1}, \ldots, h_{T_n}; \hat{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n}$$

**Result:** bond distances and bond angles.

**Challenges:**

1) Equivariant prediction with respect to any permutation of neighbors:
   $$f(h_{T_{\pi(1)}}, \ldots, h_{T_{\pi(n)}}, \hat{h}_X) = (p_{\pi(1)}, \ldots, p_{\pi(n)}), \forall \pi \in S_n$$

2) Bond distances should match,
   - i.e. $XT_1$ should have the same length predicted from the LS of $X$ or from the LS of $T_1$

3) Should explicitly address chirality (later)
Local Structure (LS) Prediction

For each non-terminal atom X, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming X is placed in 0:

\[ f(h_{T_1}, \ldots, h_{T_n}, \hat{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n} \]

**Result:** bond distances and bond angles.

**Challenges:**

1) Equivariant prediction with respect to any permutation of neighbors:

\[ f(h_{T_{\pi(1)}}, \ldots, h_{T_{\pi(n)}}, \hat{h}_X) = (p_{\pi(1)}, \ldots, p_{\pi(n)}), \forall \pi \in S_n \]

Solution: f() is a transformer w/o positional encodings taking inputs \( \{concat(h_{T_i}, h_X)\}_{i \in [1..n]} \) in any order
Local Structure (LS) Prediction

For each non-terminal atom $X$, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming $X$ is placed in $0$:

$$f(h_{T_1}, \ldots, h_{T_n}; \hat{h}_X) = (p_1, \ldots, p_n) \in \mathbb{R}^{3 \times n}$$

**Result:** bond distances and bond angles.

**Challenges:**

1) Equivariant prediction with respect to any permutation of neighbors: $f(h_{T_{\pi(1)}}, \ldots, h_{T_{\pi(n)}}; h_X) = (p_{\pi(1)}, \ldots, p_{\pi(n)}), \forall \pi \in S_n$

2) Bond distances should match, i.e. $XT_1$ should have the same length predicted from the LS of $X$ or from the LS of $T_1$

Solution: separate distance prediction from direction prediction:

$$p'_i \overset{\text{def}}{=} \frac{p_i}{\|p_i\|} d_{GNN}(h_X, h_{T_i})$$, where $d_{GNN}(h_X, h_Y) \overset{\text{def}}{=} \text{softplus}(\psi(h_X, h_Y) + \psi(h_Y, h_X)), \forall (X, Y) \in E$
Tackling Chirality

- Chiral information: Bond annotations to describe different molecules with same 2D graph, but different 3D structures (and, thus, chemical behavior).
- Reason: 3D reflection changes the molecule completely.

(a) Bromochlorofluoromethane

(b) Dichlorofluoromethane
Tackling Chirality

• Chiral information: Bond annotations to describe different molecules with same 2D graph, but different 3D structures (and, thus, chemical behavior)

(a) Bromochlorofluoromethane

(b) Dichlorofluoromethane

• Bond annotations are not fixed, i.e. multiple equivalent annotations
Tackling Chirality

• MPNNs cannot distinguish chirality …
  • … unless order of graph neighbors is explicitly used
Tackling Chirality - Our Solution

- Given a chiral center (or any center with 4 neighbors), we can compute the oriented volume.

\[
OV(p_1, p_2, p_3, p_4) \overset{\text{def}}{=} \text{sign} \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{pmatrix}
\]

- The sign of the oriented volume changes depending on chirality.

- If we get the incorrect sign, we simply reflect the structure by flipping against the z-axis.

- This ensures that all chiral centers are generated exactly, and no iterative optimization is necessary as with traditional DG-based generators. Note: DG models cannot directly solve chirality and same trick cannot be applied w/o changing the distance matrix significantly.

\[OV(C_S) = -1 \quad \text{and} \quad OV(C_R) = +1\]
Note: so far, rotation was not fixed for predicted local structures (still a degree of freedom)!

Assembling Two Neighboring LS via Torsion Angle Prediction
Assemble every two local neighboring structures.

**Predict**: torsion angles, 3-hop distances

### Assembling Two Neighboring LS via Torsion Angle Prediction
Assemble every two local neighboring structures.

**Predict:** torsion angles, 3-hop distances

**Challenges:**
- 1) Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)

  All torsions angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion:

  $$\angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_l) + \angle(XYT_i, XYZ_k) + \angle(XYZ_l, XYZ_j)] \pmod{2\pi}$$
Assemble every two local neighboring structures.

**Predict**: torsion angles, 3-hop distances

**Challenges**:
1) Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)
   - All torsions angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion:
     $$\angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_i) + \angle(XYT_i, XYZ_k) + \angle(XYZ_i, XYZ_j)] \pmod{2\pi}$$

2) Torsion angles should be predicted in a rotation-translation invariant manner, plus invariant to any permutation of $T_i$'s or of $Z_j$'s.
Assemble every two local neighboring structures.

**Predict:** torsion angles, 3-hop distances

**Challenges:**
- 1) Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)
  - All torsions angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion:
    \[ \angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_i) + \angle(XYT_i, XYZ_k) + \angle(XYZ_i, XYZ_j)](\text{mod } 2\pi) \]
- 2) Torsion angles should be predicted in a rotation-translation invariant manner, plus invariant to any permutation of $T_i$'s or of $Z_j$'s.

**Solution** - sketch (details in our paper):

Let $\Delta_{ij} \overset{\text{def}}{=} \angle(XYT_i, XYZ_j)$ and $s_{ij} \overset{\text{def}}{=} \begin{bmatrix} \cos(\Delta_{ij}) \\ \sin(\Delta_{ij}) \end{bmatrix}$
Assemble every two local neighboring structures.

**Predict:** torsion angles, 3-hop distances

**Challenges:**

1) Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)

   \[ \angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_i) + \angle(XYT_i, XYZ_k) + \angle(XYZ_i, XYZ_j)] \mod 2\pi \]

2) Torsion angles should be predicted in a rotation-translation invariant manner, plus invariant to any permutation of \( T_i \)'s or of \( Z_j \)'s.

**Solution** - sketch (details in our paper):

Let \( \Delta_{ij} \) be such that \( s := \sum_{i,j} c_{ij} \neq 0 \) and \( c_{ij} \) is defined as:

\[ c_{ij} = \frac{\cos(\Delta_{ij})}{\sin(\Delta_{ij})} \]

Let fixed \( c_{ij} \in \mathbb{R} \) such that \( s := \sum_{i,j} c_{ij} \neq 0 \), e.g. \( c_{ij} = MLP(h_{T_i} + h_{Z_j}) \)
Assemble every two local neighboring structures.

**Predict:** torsion angles, 3-hop distances

**Challenges:**

1. Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)
   
   All torsions angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion:
   
   $\angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_i) + \angle(XYT_i, XYZ_k) + \angle(XYZ_i, XY Z_j)](\text{mod } 2\pi)$

2. Torsion angles should be predicted in a rotation-translation invariant manner, plus invariant to any permutation of $T_i$'s or of $Z_j$'s.

**Solution** - sketch (details in our paper):

Let $\Delta_{ij} \equiv \angle(XYT_i, XYZ_j)$ and $s_{ij} \equiv \begin{bmatrix} \cos(\Delta_{ij}) \\ \sin(\Delta_{ij}) \end{bmatrix}$

Let fixed $c_{ij} \in \mathbb{R}$ such that $s := \sum_{i,j} c_{ij}s_{ij} \neq 0$, e.g. $c_{ij} = MLP(h_{i_i} + h_{Z_j})$

Then, the following definition gives an **unique** $\alpha$ per each true torsion angle:

$$\begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \equiv \frac{s^*}{\|s^*\|}$$
Assemble every two local neighboring structures.

**Predict**: torsion angles, 3-hop distances

**Challenges**:

1) Parameterize single canonical torsion angle per rotatable bond (no over-parameterization)

   \[ \angle(XYT_i, XYZ_j) = \frac{\cos(\Delta_{ij})}{\sin(\Delta_{ij})} \]

2) Torsion angles should be predicted in a rotation-translation invariant manner, plus invariant to any permutation of \(T_i\)'s or of \(Z_j\)'s.

**Solution** - sketch (details in our paper):

Let \( \Delta_{ij} \) and \( s_{ij} \) such that \( s := \sum_{i,j} c_{ij} s_{ij} \neq 0 \), e.g. \( c_{ij} = MLP(h_{T_i} + h_Z) \)

Then, the following definition gives an unique \( \alpha \) per each true torsion angle:

\[
\begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \overset{\text{def}}{=} \frac{s^*}{\|s^*\|}
\]

so it can be predicted using a symmetric NN (to ensure consistency w.r.t. X and Y):

\[
\alpha = [\phi(h_X, h_Y, h_{mol}) + \phi(h_Y, h_X, h_{mol})] \mod 2\pi
\]
Optimal Transport Loss

- First setting: one predicted conformer and one ground truth

\[ \mathcal{L}(C, C^*) \overset{\text{def}}{=} \xi_1 \cdot \frac{1}{\#\{(u, v) \in E\}} \sum_{\{(u, v) \in E\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_2 \cdot \frac{1}{\#\{(u, v) : 2\text{-hops away}\}} \sum_{\{u, v : 2\text{-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_3 \cdot \frac{1}{\#\{(u, v) : 3\text{-hops away}\}} \sum_{\{u, v : 3\text{-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ - \xi_4 \cdot \frac{1}{\#(u, v), (v, w) \in E, (v, w) \in E} \sum_{(u, v) \in E, (v, w) \in E} \cos(\angle uvw - \angle^* uvw) \]

\[ - \xi_5 \cdot \frac{1}{\#(u, v), (v, w), (w, y) \in E} \sum_{(u, v), (v, w), (w, y) \in E} \cos(\angle uvw, vwy) - \angle^*(uvw, vwy) \]
Optimal Transport Loss

• First setting: one predicted conformer and one ground truth

\[ \mathcal{L}(C, C^*) \overset{\text{def}}{=} \xi_1 \cdot \frac{1}{\#\{(u, v) \in E\}} \sum_{\{(u, v) \in E\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_2 \cdot \frac{1}{\#\{u, v : 2\text{-hops away}\}} \sum_{\{u, v : 2\text{-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_3 \cdot \frac{1}{\#\{u, v : 3\text{-hops away}\}} \sum_{\{u, v : 3\text{-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ - \xi_4 \cdot \frac{1}{\#(u, v) \in E, (v, w) \in E} \sum_{(u, v) \in E, (v, w) \in E} \cos(\angle uvw - \angle^* uvw) \]

\[ - \xi_5 \cdot \frac{1}{\#(u, v), (v, w), (w, y) \in E} \sum_{(u, v), (v, w), (w, y) \in E} \cos(\angle uvw, vwz) - \angle^* (uvw, vwy) \]

• Hard case: Multiple (variable sized) ground truth conformers \{C_1^*, \ldots, C_n^*\} and predicted \{C_1, \ldots, C_m\}

• How to leverage the single conformer loss \(\mathcal{L}(C, C^*)\) ?

• How to avoid adversarial training (impractical, hard to train, etc) ?

• How to generate diverse conformers (to cover all modes of the true distribution) ?
Optimal Transport Loss

• First setting: one predicted conformer and one ground truth

\[ \mathcal{L}(C, C^*) = \xi_1 \cdot \frac{1}{\# \{(u, v) \in E\}} \sum_{\{(u, v) \in E\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_2 \cdot \frac{1}{\# \{(u, v : 2\text{-hops away})\}} \sum_{\{u, \text{: 2-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ + \xi_3 \cdot \frac{1}{\# \{(u, v : 3\text{-hops away})\}} \sum_{\{u, \text{: 3-hops away}\}} (d(u, v) - d^*(u, v))^2 \]

\[ - \xi_4 \cdot \frac{1}{\# \{(u, v) \in E, (v, w) \in E\}} \sum_{\{(u, v) \in E, (v, w) \in E\}} \cos(\angle uvw - \angle^* uvw) \]

\[ - \xi_5 \cdot \frac{1}{\# \{(u, v), (v, w), (w, y) \in E\}} \sum_{\{(u, v), (v, w), (w, y) \in E\}} \cos(\angle uvw, vw, wvy) - \angle^* (uvw, wvy) \]

• Hard case: Multiple (variable sized) ground truth conformers \( \{C_1^*, \ldots, C_n^*\} \) and predicted \( \{C_1, \ldots, C_m\} \)

How to leverage the single conformer loss \( \mathcal{L}(C, C^*) \)?

How to avoid adversarial training (impractical, hard to train, etc)?

How to generate diverse conformers (to cover all modes of the true distribution)?

Solution: Wasserstein loss: \( \mathcal{L} = \mathcal{W}_{\mathcal{L}(\cdot, \cdot)}(\{C_j\}_j, \{C_i^*\}_i) = \inf_{T \in \mathbb{R}^{m \times n}} \sum_{i=1}^{n} \sum_{j=1}^{m} T_{ij} \mathcal{L}(C_j, C_i^*) \) where \( T \) is doubly stochastic.
Dealing with Symmetry

- A & B are identical and symmetric in the molecular graph, but not in 3D (since $d(A, C) \neq d(B, C)$)
Dealing with Symmetry

- A & B are identical and symmetric in the molecular graph, but not in 3D (since $d(A, C) \neq d(B, C)$)
- We use a loss that tries all permutations of such symmetric groups of terminal atoms:

\[
\mathcal{L}_{\text{perm}}(C, C^*) := \min \left( \mathcal{L}(C_{A,B}, C_{A,B}^*), \mathcal{L}(C_{B,A}, C_{A,B}^*) \right)
\]
Assemble Full Conformer at Test Time

- We smooth ring structures by systematically beginning the assembly process at different atoms; this creates $n$ structures for a ring with $n$ edges.
- We add atoms in succession until we close the ring.
- Finally, we average all $n$ structures, which results in the final smoothed ring.

![Image of chemical structures comparing ground truth, model without ring correction, and model with ring correction.]
Datasets

- **Datasets:**
  - QM9 conformers: small molecules
  - DRUGS (Axelrod and Gomez-Bombarelli, 2020): larger drug-like molecules
  - Conformers generated with semi-empirical tight-binding DFT (GFN2-xTB) generated with the CREST software
Evaluation Metrics

• Comparison metric of two conformers: **RMSD** (Root-mean-square deviation of atomic positions):
  - RMSD compares any two point clouds \( p_1, \ldots, p_n \in \mathbb{R}^3 \) and \( q_1, \ldots, q_n \in \mathbb{R}^3 \)
  - First, they are aligned using the best rotation and translation: \( H^*, b^* = \arg \min_{H \in O(3), b \in \mathbb{R}^3} \sum_{k=1}^{n} ||p_k - Hq_k - b||^2 \)
  - Finally, standard RMSE is computed: \( RMSD(p, q) := \sqrt{\frac{1}{n} \sum_{k=1}^{n} ||p_k - H^*q_k - b^*||^2} \)

• Comparison of two conformer distributions:
  - **Coverage** \( \uparrow \): percentage of “correct” generated conformers from ground truth set.
    \[
    \text{COV - R (Recall)} \overset{\text{def}}{=} \frac{1}{L} |\{ l \in [1..L] : \exists k \in [1..K], RMSD(C_k, C^*_l) < \delta \}| 
    \]
  - **Average Minimum RMSD** \( \downarrow \): for each generated conformer, compute RMSD to the closest ground truth. Average over all.
    \[
    \text{AMR - R (Recall)} \overset{\text{def}}{=} \frac{1}{L} \sum_{l \in [1..L]} \min_{k \in [1..K]} RMSD(C_k, C^*_l) 
    \]
  - COV - R (Precision) and AMR - R (Precision) defined similarly
For each molecule, we ask models to generate 2x as many conformers as in the ground truth

Test set: 1000 molecules

<table>
<thead>
<tr>
<th>Models</th>
<th>COV - R (%) ↑</th>
<th>AMR - R (Å) ↓</th>
<th>COV - P (%) ↑</th>
<th>AMR - P (Å) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>GraphDG (ML)</td>
<td>10.37</td>
<td>0.00</td>
<td>1.950</td>
<td>1.933</td>
</tr>
<tr>
<td>CGCF (ML)</td>
<td>54.35</td>
<td>56.74</td>
<td>1.248</td>
<td>1.224</td>
</tr>
<tr>
<td>RDKit/ETKDG</td>
<td>68.78</td>
<td>76.04</td>
<td>1.042</td>
<td>0.982</td>
</tr>
<tr>
<td>OMEGA (C)</td>
<td>81.64</td>
<td>97.25</td>
<td>0.851</td>
<td>0.771</td>
</tr>
<tr>
<td>GEO MOL (s = 9.5)</td>
<td><strong>86.07</strong></td>
<td><strong>98.06</strong></td>
<td><strong>0.846</strong></td>
<td><strong>0.820</strong></td>
</tr>
<tr>
<td>GEO MOL (s = 5)</td>
<td>82.43</td>
<td>95.10</td>
<td>0.862</td>
<td>0.837</td>
</tr>
</tbody>
</table>

Table 2: Results on the GEOM-QM9 dataset. See caption of table 1.
Num of rotatable bonds vs coverage
Recall vs Threshold (no FF)
<table>
<thead>
<tr>
<th>Reference</th>
<th>GraphDG</th>
<th>CGCF</th>
<th>ETKDG</th>
<th>OMEGA</th>
<th>GeoMol</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image25.png" alt="Image" /></td>
<td><img src="image26.png" alt="Image" /></td>
<td><img src="image27.png" alt="Image" /></td>
<td><img src="image28.png" alt="Image" /></td>
<td><img src="image29.png" alt="Image" /></td>
<td><img src="image30.png" alt="Image" /></td>
</tr>
<tr>
<td>Reference</td>
<td>GraphDG</td>
<td>CGCF</td>
<td>ETKDG</td>
<td>OMEGA</td>
<td>GeoMol</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td><img src="reference1.png" alt="Image" /></td>
<td><img src="graphdg1.png" alt="Image" /></td>
<td><img src="cgcf1.png" alt="Image" /></td>
<td><img src="etkdg1.png" alt="Image" /></td>
<td><img src="omega1.png" alt="Image" /></td>
<td><img src="geomol1.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="reference2.png" alt="Image" /></td>
<td><img src="graphdg2.png" alt="Image" /></td>
<td><img src="cgcf2.png" alt="Image" /></td>
<td><img src="etkdg2.png" alt="Image" /></td>
<td><img src="omega2.png" alt="Image" /></td>
<td><img src="geomol2.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="reference3.png" alt="Image" /></td>
<td><img src="graphdg3.png" alt="Image" /></td>
<td><img src="cgcf3.png" alt="Image" /></td>
<td><img src="etkdg3.png" alt="Image" /></td>
<td><img src="omega3.png" alt="Image" /></td>
<td><img src="geomol3.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="reference4.png" alt="Image" /></td>
<td><img src="graphdg4.png" alt="Image" /></td>
<td><img src="cgcf4.png" alt="Image" /></td>
<td><img src="etkdg4.png" alt="Image" /></td>
<td><img src="omega4.png" alt="Image" /></td>
<td><img src="geomol4.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Conformer Generation Time

![Graph showing time per conformer vs number of rotatable bonds for different methods: GraphDG, CGCF, ETKDG, OMEGA, GeoMol. The x-axis represents the number of rotatable bonds, and the y-axis represents the time per conformer in seconds.](image)

- GraphDG
- CGCF
- ETKDG
- OMEGA
- GeoMol
### Results (w/ FF fine-tuning)

Table 6: Results on the GEOM-DRUGS dataset with FF fine-tuning.

<table>
<thead>
<tr>
<th>Models</th>
<th>COV - R (%) ↑</th>
<th>AMR - R (Å) ↓</th>
<th>COV - P (%) ↑</th>
<th>AMR - P (Å) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>GraphDG (ML)</td>
<td>85.33</td>
<td>100.00</td>
<td>0.859</td>
<td>0.831</td>
</tr>
<tr>
<td>CGCF (ML)</td>
<td>91.25</td>
<td><strong>100.00</strong></td>
<td>0.723</td>
<td>0.702</td>
</tr>
<tr>
<td>RDKit/ETKDG</td>
<td>77.22</td>
<td>87.62</td>
<td>0.886</td>
<td>0.837</td>
</tr>
<tr>
<td>OMEGA (C)</td>
<td>80.82</td>
<td>94.74</td>
<td>0.840</td>
<td>0.755</td>
</tr>
<tr>
<td>GeoMol (s = 9.5)</td>
<td>89.83</td>
<td>99.57</td>
<td>0.706</td>
<td>0.685</td>
</tr>
<tr>
<td>GeoMol (s = 3.0)</td>
<td><strong>91.34</strong></td>
<td><strong>100.00</strong></td>
<td><strong>0.683</strong></td>
<td><strong>0.663</strong></td>
</tr>
</tbody>
</table>

Table 7: Results on the GEOM-QM9 dataset with FF fine-tuning.

<table>
<thead>
<tr>
<th>Models</th>
<th>COV - R (%) ↑</th>
<th>AMR - R (Å) ↓</th>
<th>COV - P (%) ↑</th>
<th>AMR - P (Å) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>GraphDG (ML)</td>
<td>88.70</td>
<td>100.00</td>
<td>0.210</td>
<td>0.165</td>
</tr>
<tr>
<td>CGCF (ML)</td>
<td>75.45</td>
<td>100.00</td>
<td>0.313</td>
<td>0.246</td>
</tr>
<tr>
<td>RDKit/ETKDG</td>
<td>83.48</td>
<td><strong>100.00</strong></td>
<td>0.219</td>
<td>0.172</td>
</tr>
<tr>
<td>OMEGA (C)</td>
<td>85.73</td>
<td><strong>100.00</strong></td>
<td><strong>0.177</strong></td>
<td><strong>0.126</strong></td>
</tr>
<tr>
<td>GeoMol (s = 5)</td>
<td><strong>89.37</strong></td>
<td><strong>100.00</strong></td>
<td>0.201</td>
<td>0.157</td>
</tr>
</tbody>
</table>
Current Limitations

- Weakness in capturing some long-range interactions especially of structures that are scarce in the train set (e.g. macrocycles)
- Steric clashes
- Large rings
Thank you!