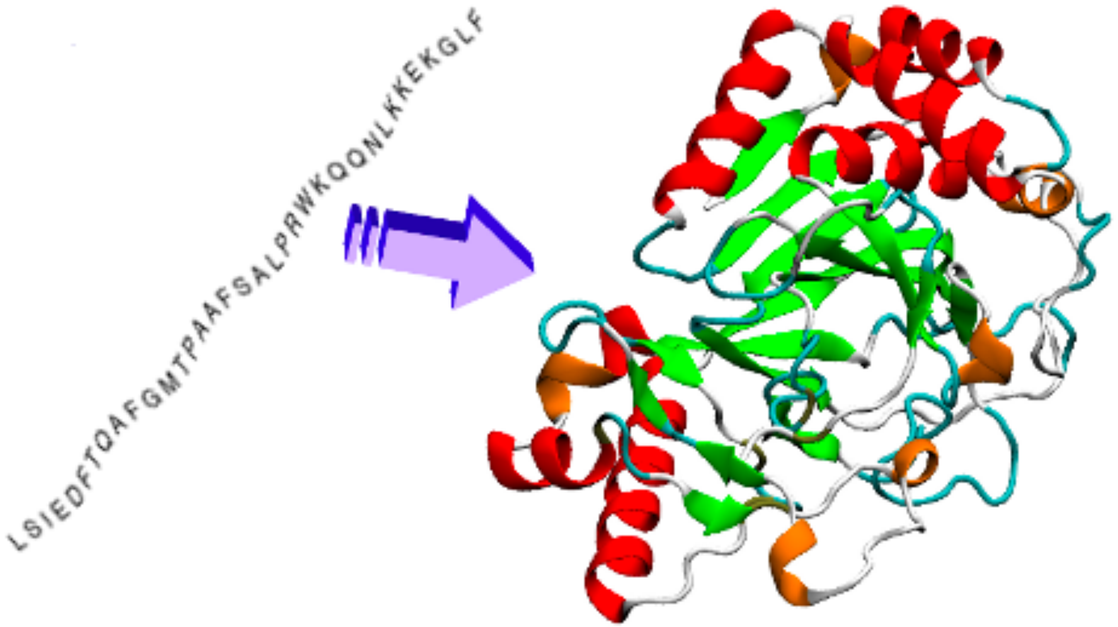
GeoMol: Torsional Geometric Generation of Molecular 3D Conformer Ensembles

Octavian Ganea*, Lagnajit Pattanaik*, Connor W. Coley, William H. Green,
Regina Barzilay, Klavs F. Jensen, and Tommi S. Jaakkola

MIT Computer Science & Artificial Intelligence Laboratory

Motivation

- Recent breakthrough in **Protein folding** (e.g AlphaFold2 and RoseTTAFold)



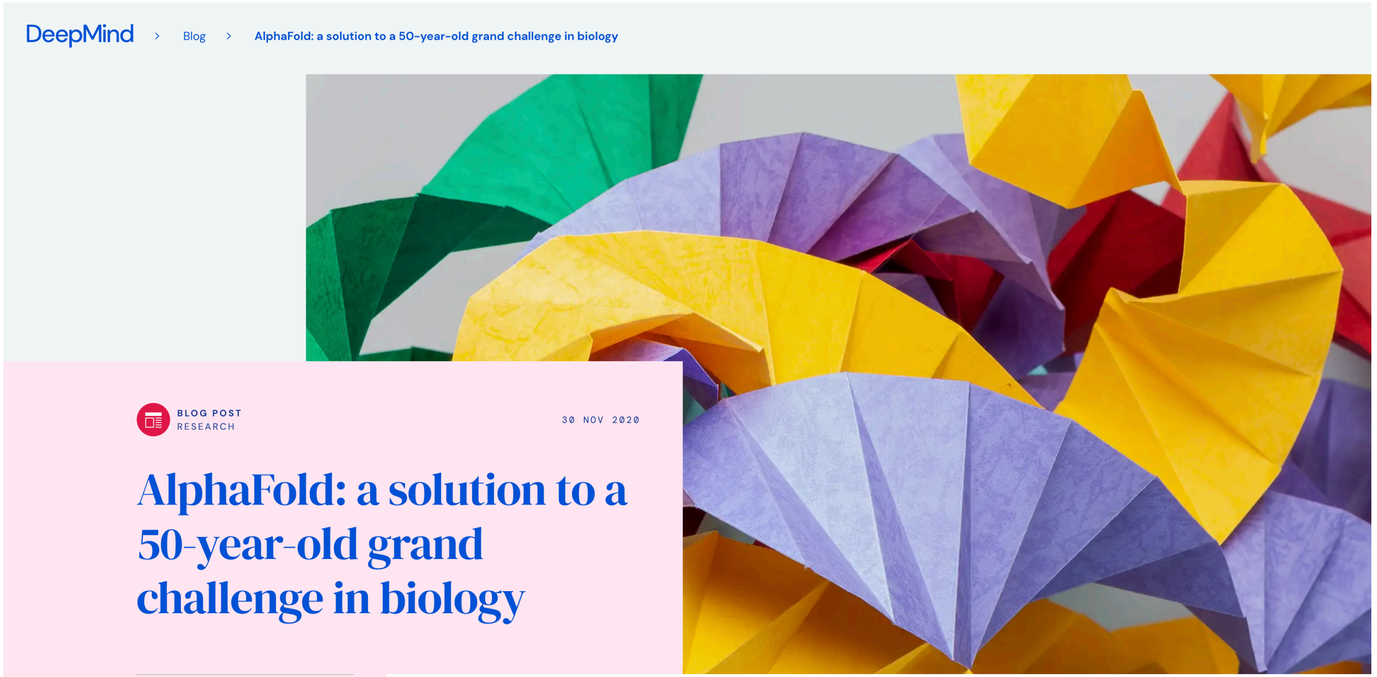
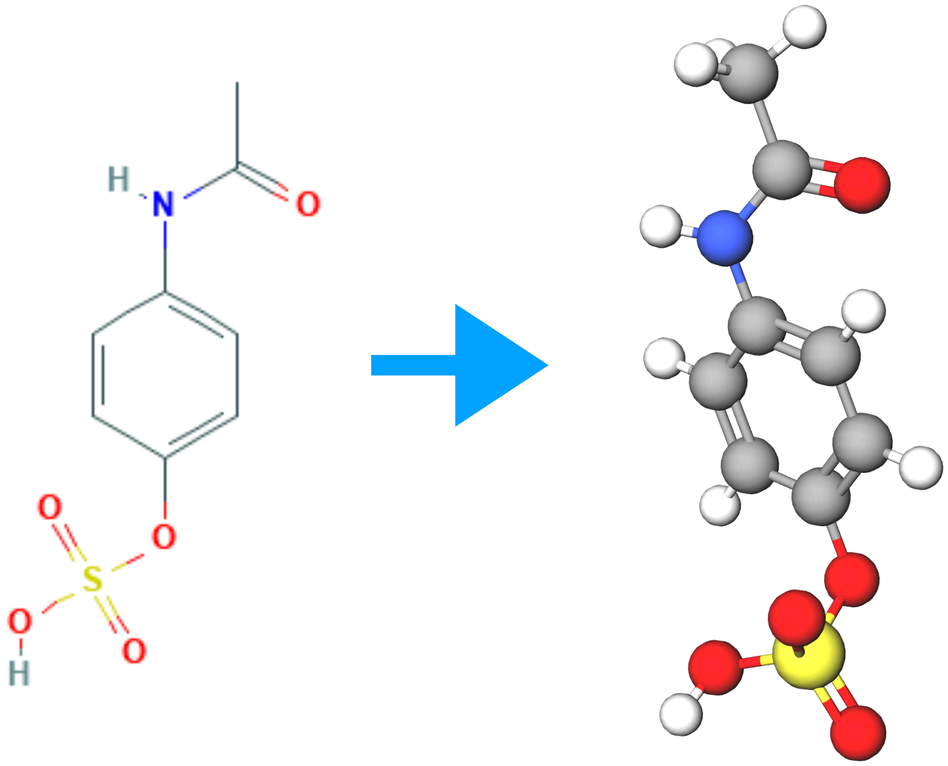


Fig source: sites.google.com/site/fabiopietrucci/ and https://deepmind.com/

Motivation

- How about molecules ? They naturally lie in the 3D space.



Motivation

- Important applications, e.g. drug - target interactions depend on the 3D structures of both the protein and the ligand.

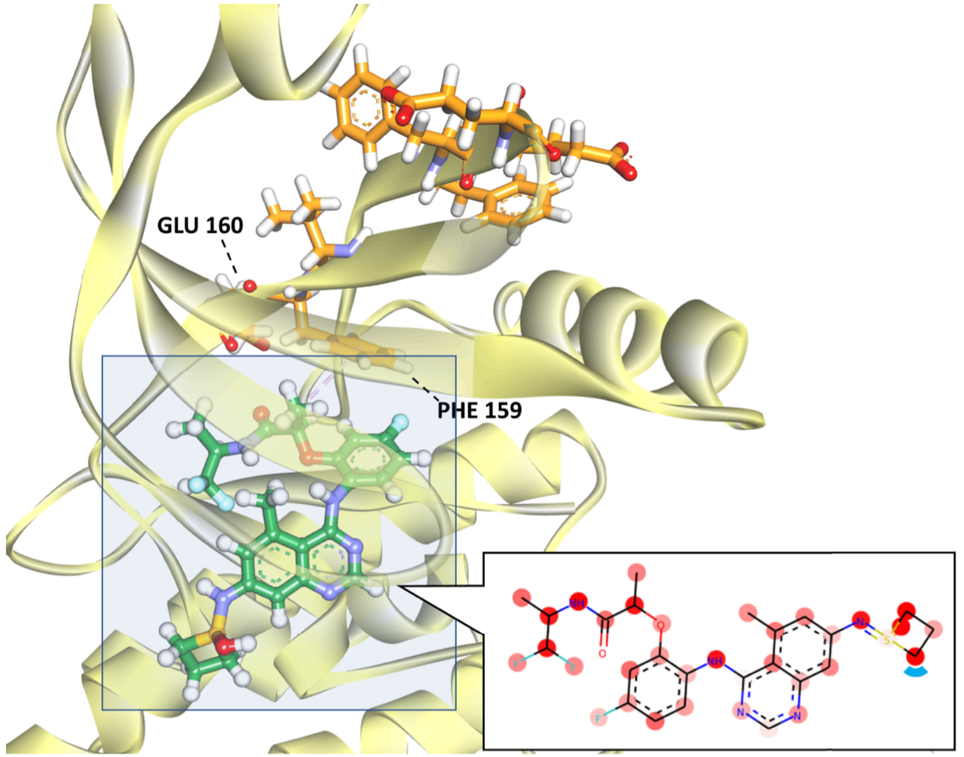
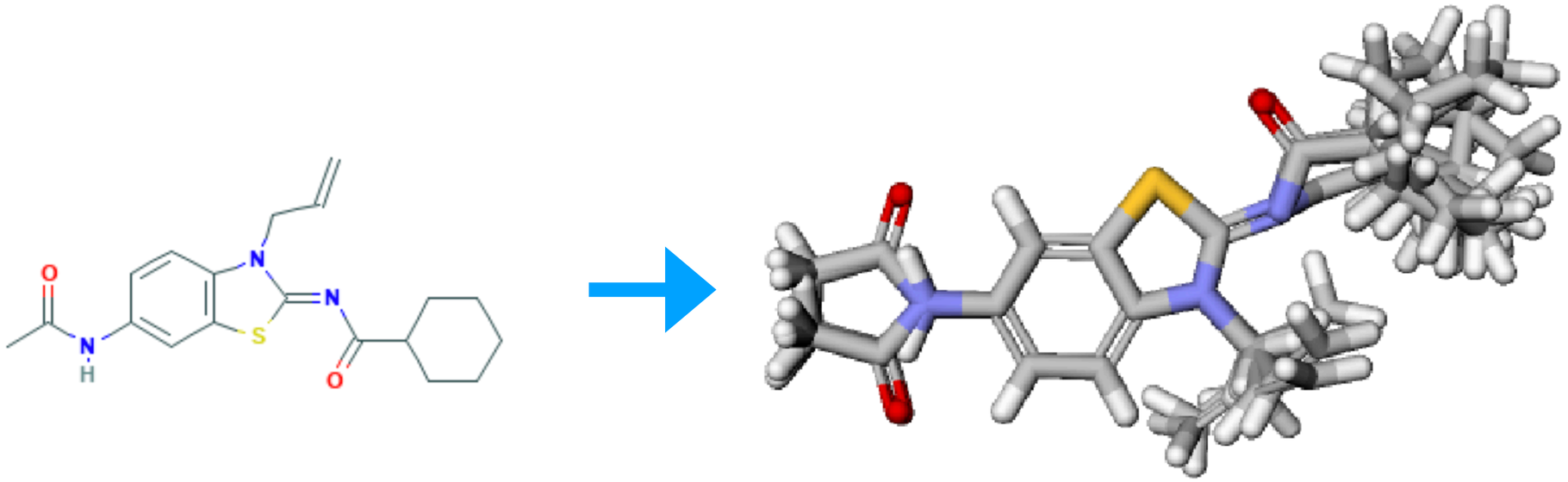


Figure 3: The interaction between SCHEMBL16362922 and the MAP kinase-interacting serine/threonine-protein kinase 2. The protein is shown in yellow and the small molecule is shown in green.

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

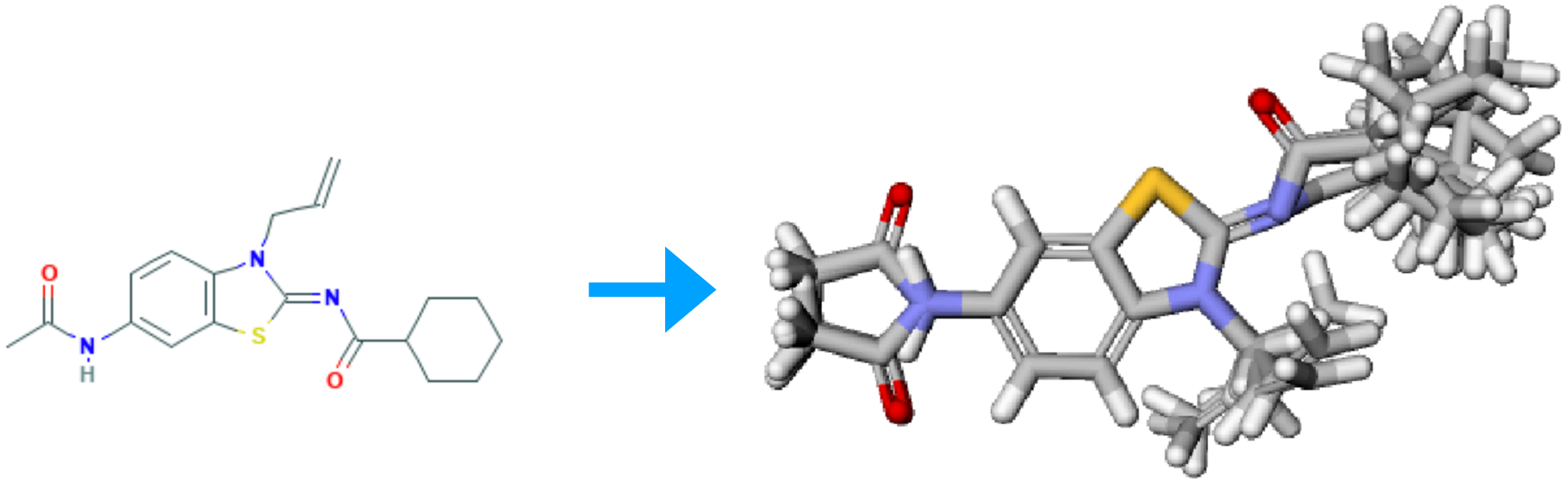
Motivation

- Molecules have potentially thousands of stable conformations.



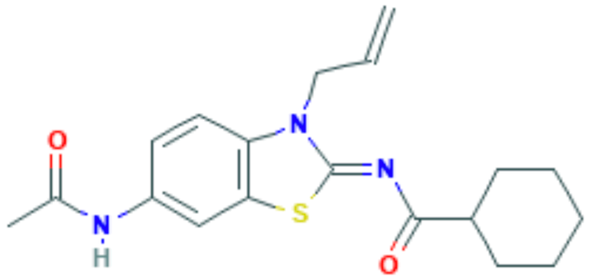
Motivation

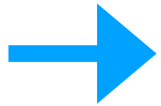
- Molecules have potentially thousands of stable conformations.
    - Model rigid and flexible parts of the 3D structure.

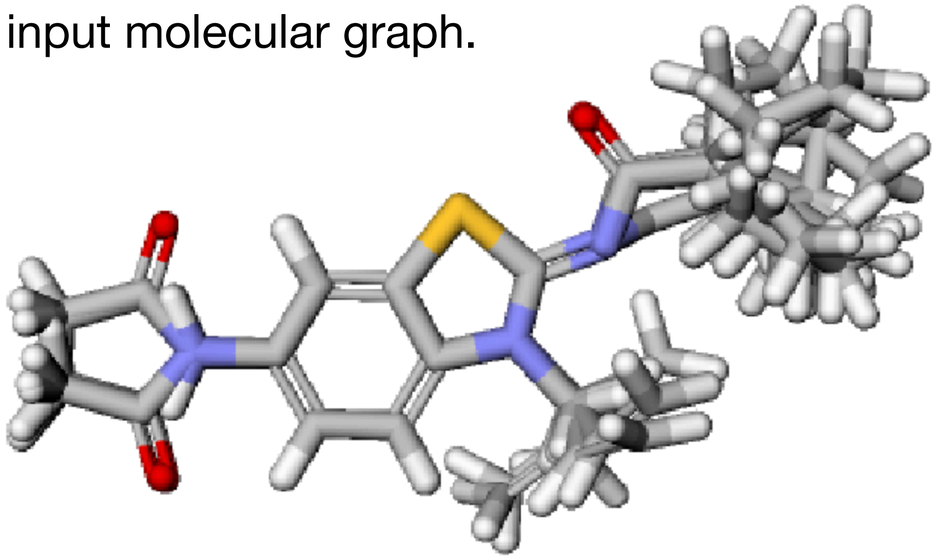


Motivation

- Molecules have potentially thousands of stable conformations.
  - Model rigid and flexible parts of the 3D structure.
- **Goal**: predict (all) low energy conformers of an input molecular graph.







Motivation

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

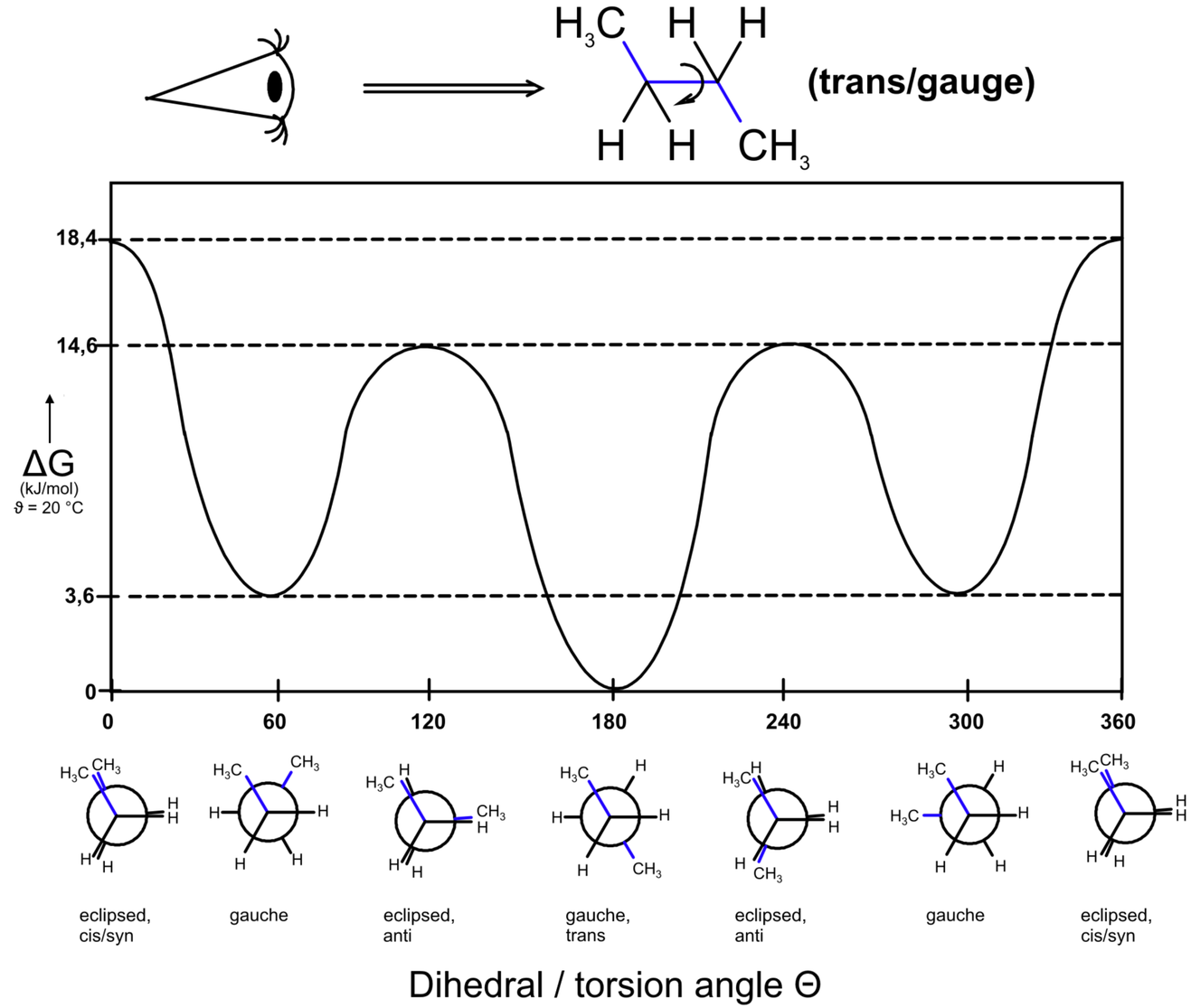


Fig source: Wikipedia

Motivation

- **Goal**: predict (all) low energy conformers of an input molecular graph

- **Why?**
  - Faster, computationally efficient and more accurate conformer 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

  - Intermediate representation for various property predictors (e.g. biological activity); pre-training technique

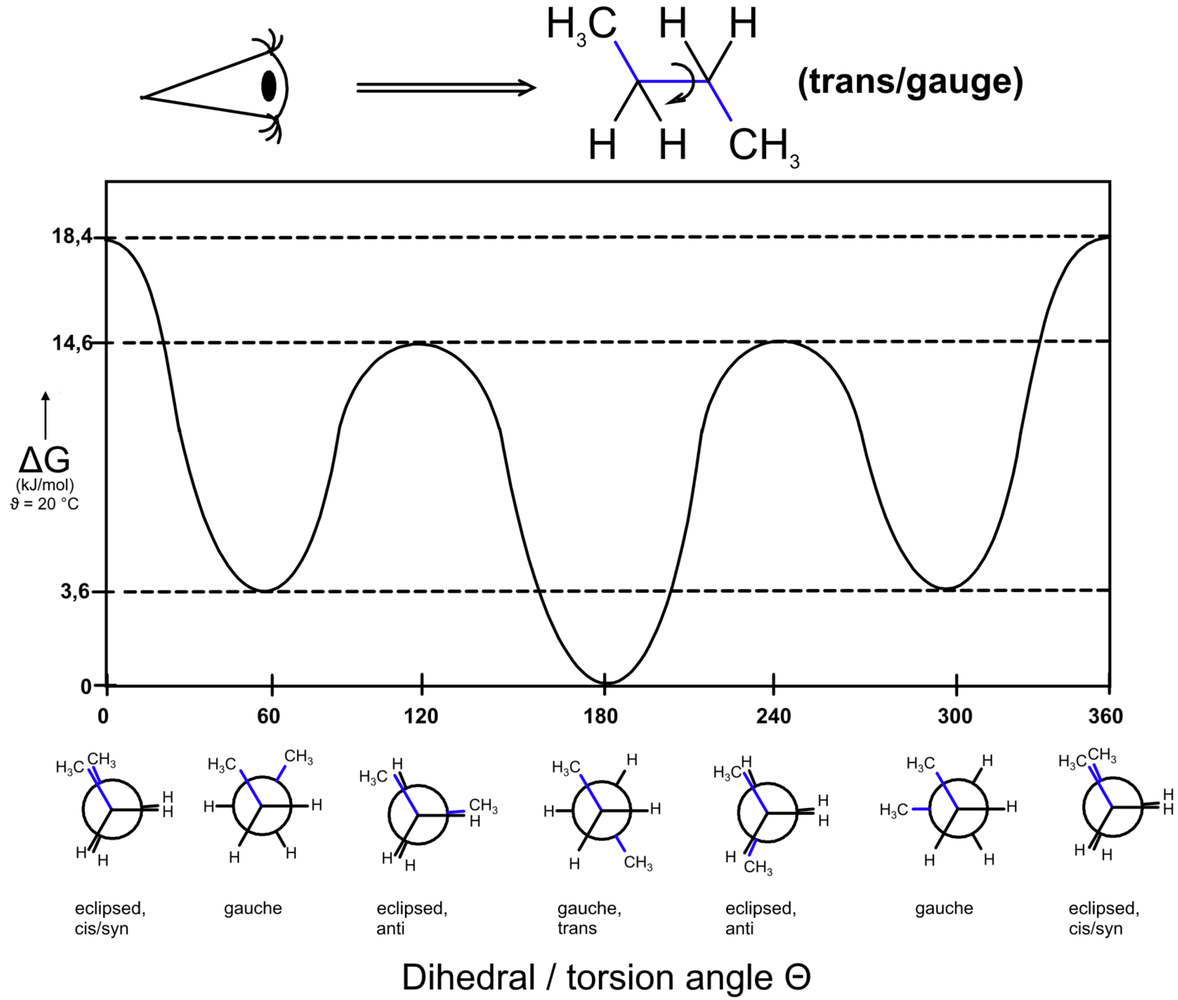


Fig source: Wikipedia

Traditional Conformer generation

- **Stochastic methods:**
  - Distance geometry initialization + subsequent optimization
  - Popular open-source method: ETKDG/RDKit

Traditional Conformer generation

- **Stochastic methods:**
  - Distance geometry initialization + subsequent optimization
  - Popular open-source method: ETKDG/RDKit

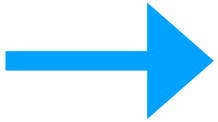
Drawbacks:

- Difficult to sample diverse and representative conformers
- Computationally expensive

Traditional Conformer generation

- **Stochastic methods:**
  - Distance geometry initialization + subsequent optimization
  - Popular open-source method: ETKDG/RDKit

- **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

Traditional Conformer generation

- **Stochastic methods:**
  - Distance geometry initialization + subsequent optimization
  - Popular open-source method: ETKDG/RDKit

Drawbacks:

- Difficult to sample diverse and representative conformers
- Computationally expensive

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

Drawbacks:

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

Traditional Conformer generation

- **Stochastic methods:**
  - Distance geometry initialization + subsequent optimization
  - Popular open-source method: ETKDG/RDKit

Drawbacks:

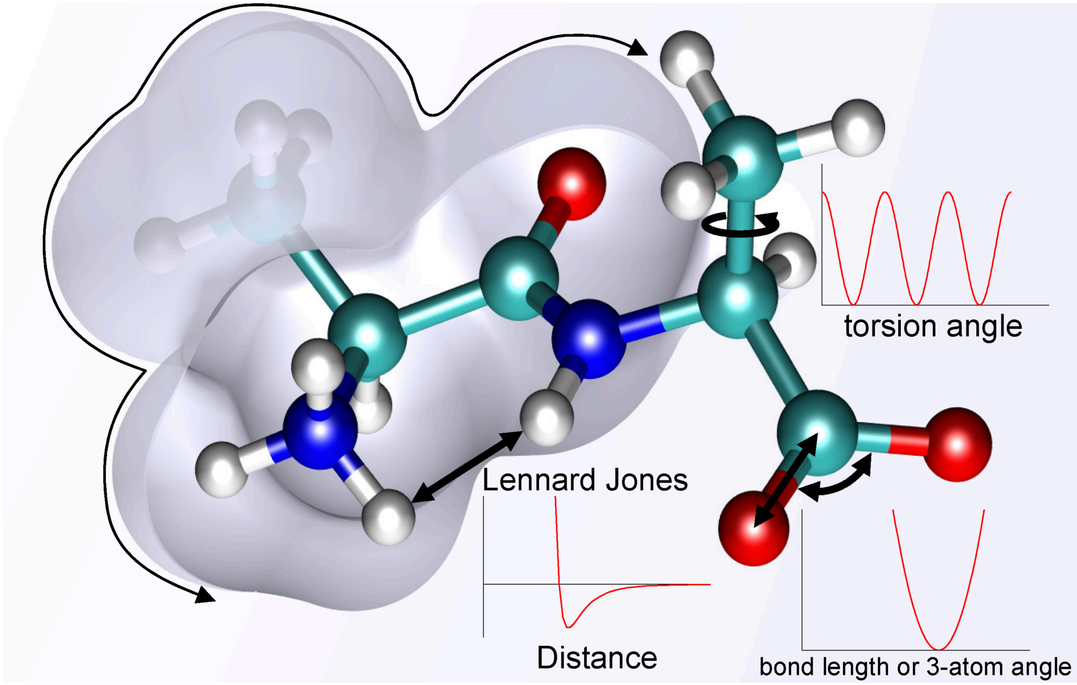
- Difficult to sample diverse and representative conformers
- Computationally expensive

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

Drawbacks:

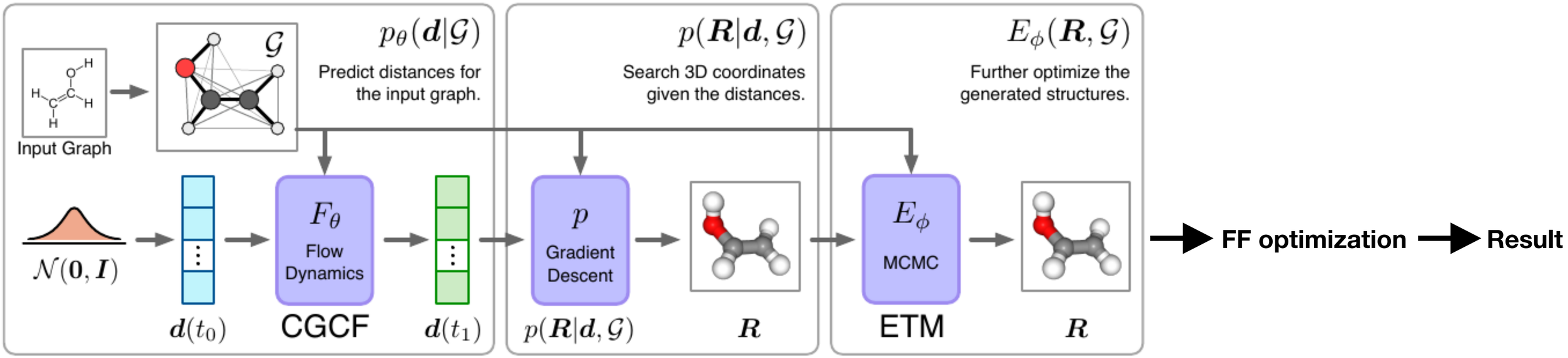
- Computational prohibitive for structures with large number of rotatable bonds
- Poor generalization to unseen structures

- Fine-tunning with **Force Fields (FF)** is often needed
  - Crude approximations of the true energy
  - Experimental quantum mechanics parameters
  - Strong assumptions (simplistic formulas)
  - Other limitations, e.g. ability to accurately capture subtle, weak interactions in biomolecules.



ML Approaches for Conformer generation

- Multi-stage models: generate distance matrix, then predict coordinates, then fine-tune the conformer [2,3]
  - Need a FF or extra energy model
  - Not trainable end-to-end ⟶ error accumulation
  - No explicit handling of classic molecular geometry: bond angles, torsions angles, chirality, cis/trans conformations , etc.
  - Requires an iterative procedure to sample conformers (e.g. via Langevin dynamics [3])

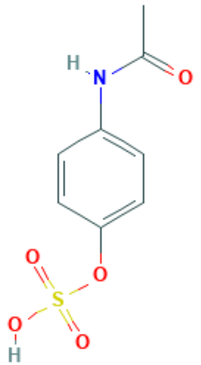


[1] Simm, Gregor NC, and José Miguel Hernández-Lobato. "A generative model for molecular distance geometry.", ICML 2020

[2] Learning neural generative dynamics for molecular conformer generation. ICLR 2021

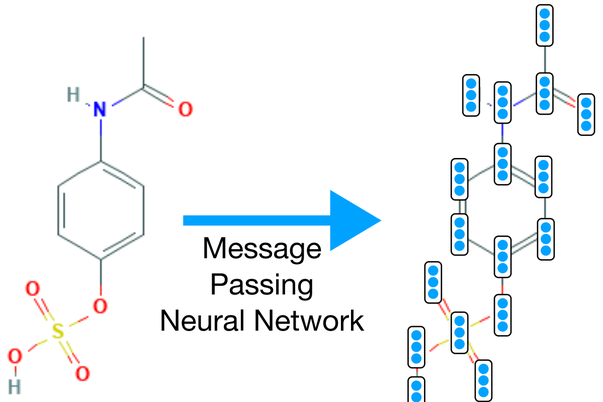
[3] Learning Gradient Fields for Molecular Conformation Generation, Shi et al, ICML 2021

GeoMol - Overview



Input:
molecular graph
(+ random noise)

GeoMol - Overview

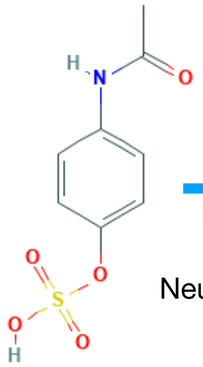


Input:
molecular graph
(+ random noise)

Message
Passing
Neural Network

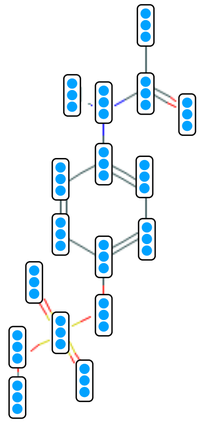
Atom
embeddings

GeoMol - Overview

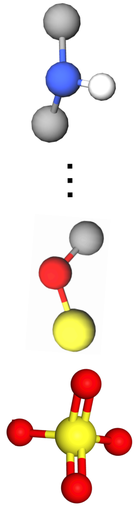


Input:
molecular graph
(+ random noise)

Message
Passing
Neural Network

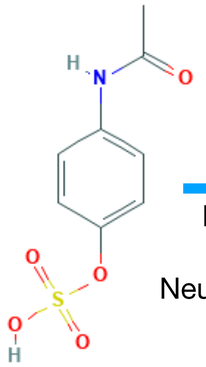


Atom
embeddings



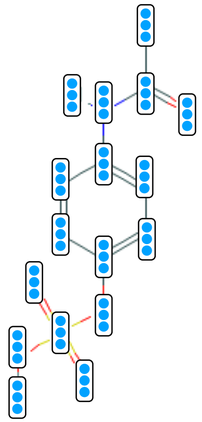
Predict **local structures**,
i.e., 1-hop 3D coordinates.
Result: bond distances and bond angles.

GeoMol - Overview

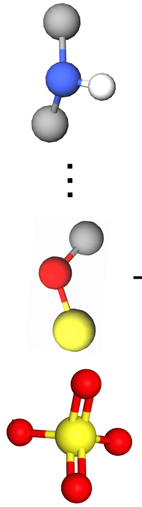


Input:
molecular graph
(+ random noise)

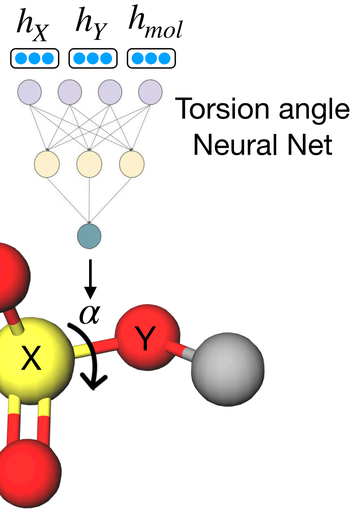
Message
Passing
Neural Network



Atom
embeddings

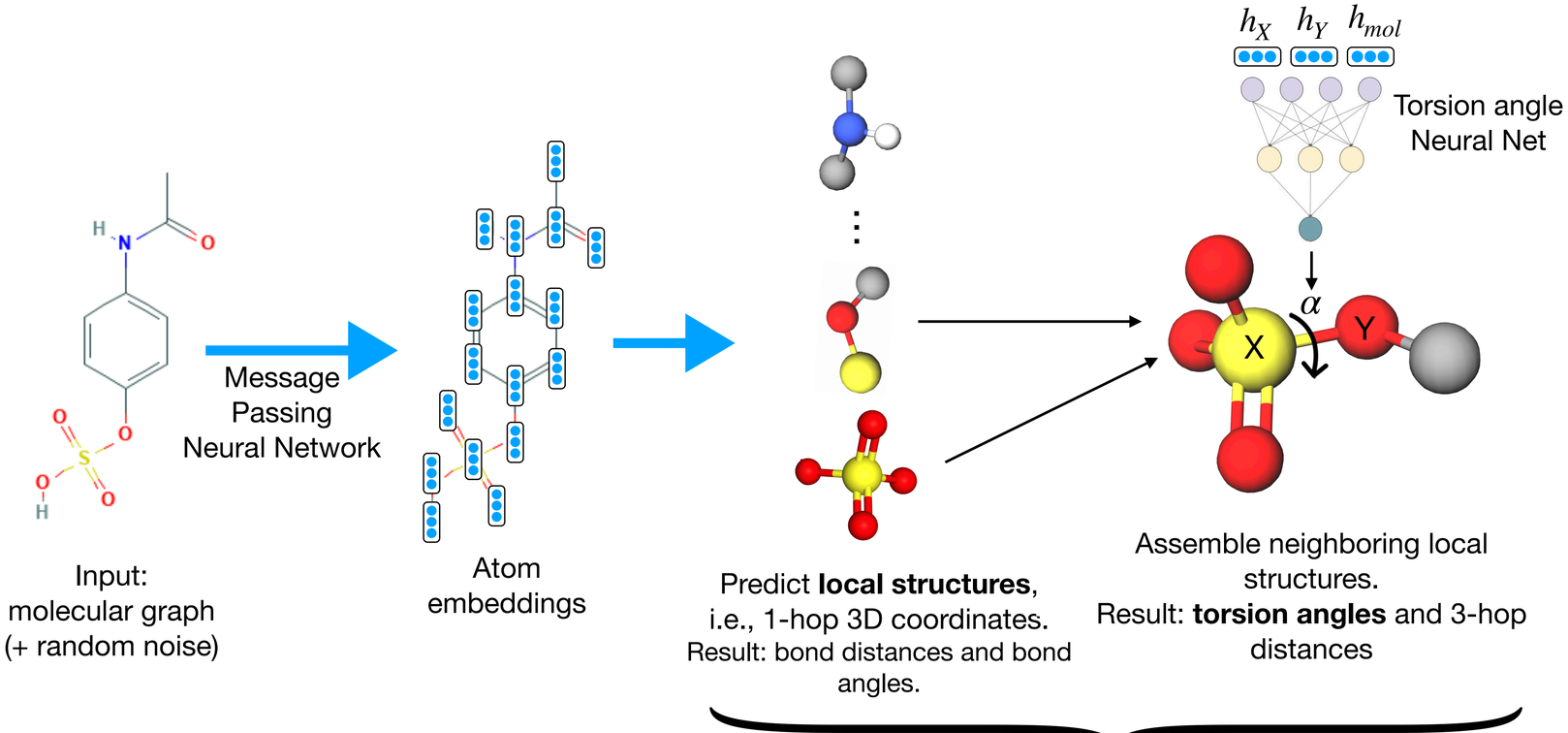


Predict **local structures**,
i.e., 1-hop 3D coordinates.
Result: bond distances and bond angles.

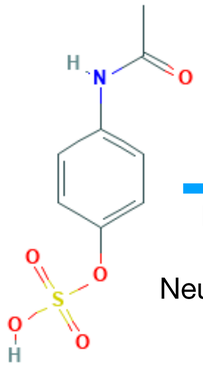


Assemble neighboring local structures.
Result: **torsion angles** and 3-hop distances

GeoMol - Overview

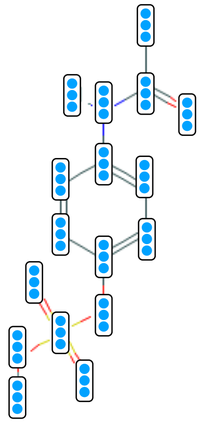


GeoMol - Overview

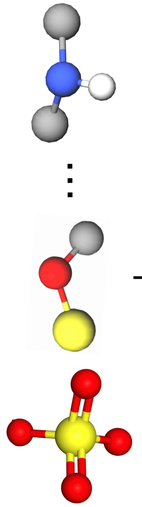


Input:
molecular graph
(+ random noise)

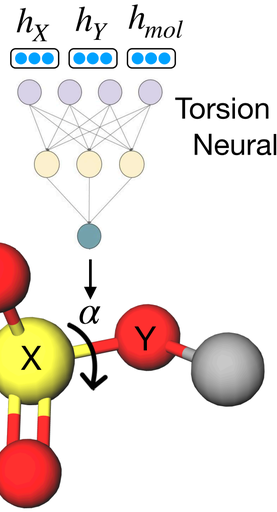
Message
Passing
Neural Network



Atom
embeddings



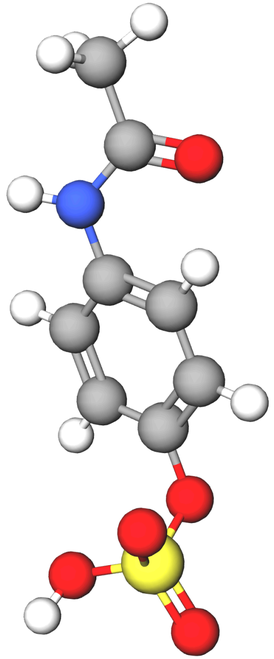
Predict **local structures**,
i.e., 1-hop 3D coordinates.
Result: bond distances and bond angles.



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

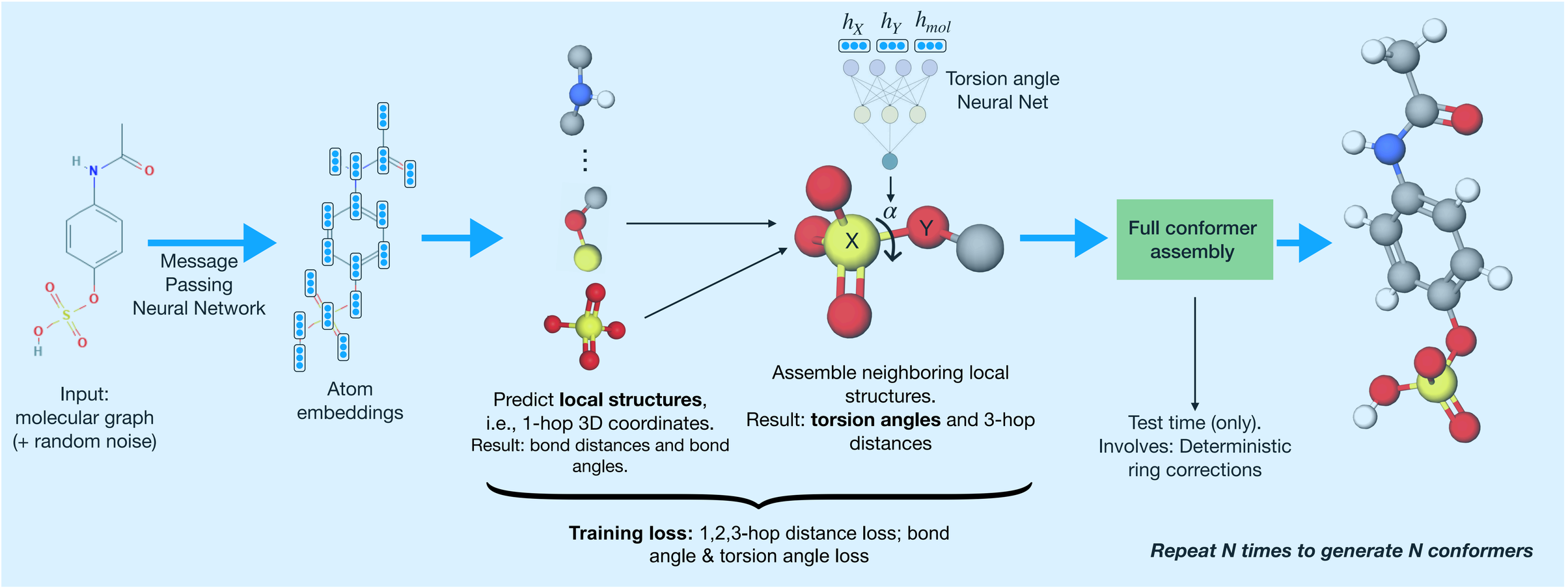
Full conformer assembly

Test time (only).
Involves: Deterministic ring corrections

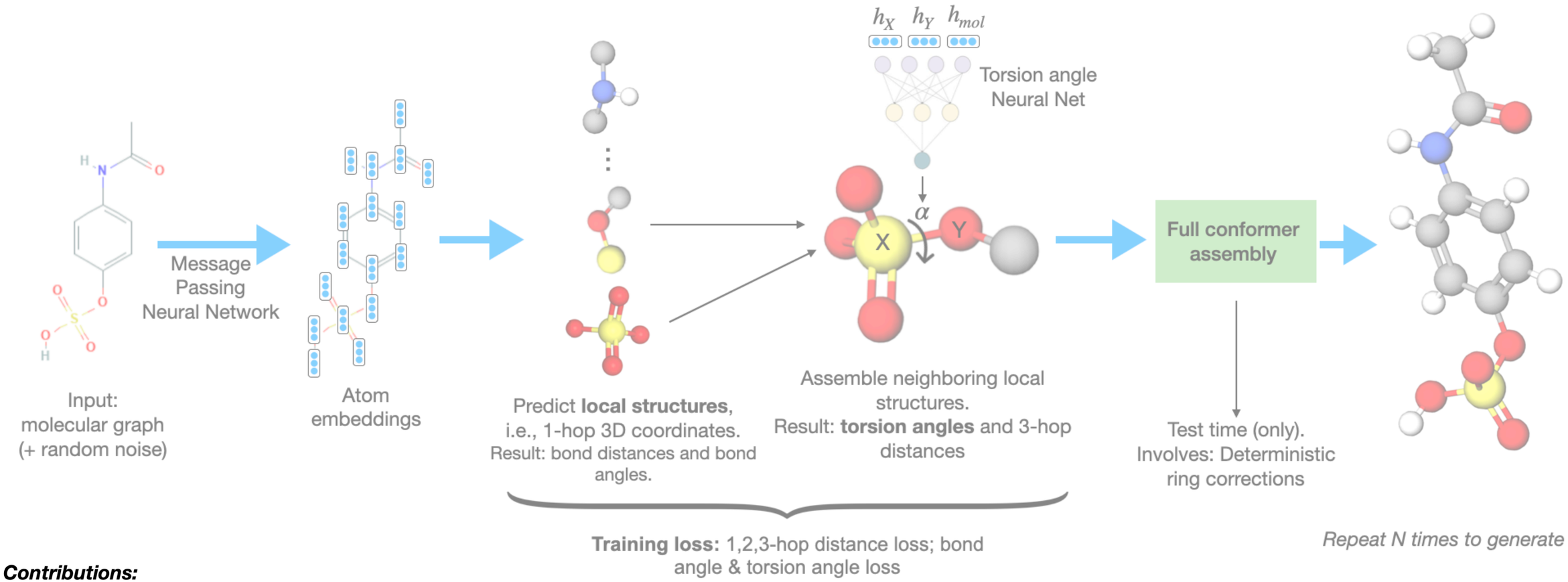


Training loss: 1,2,3-hop distance loss; bond angle & torsion angle loss

GeoMol - Overview



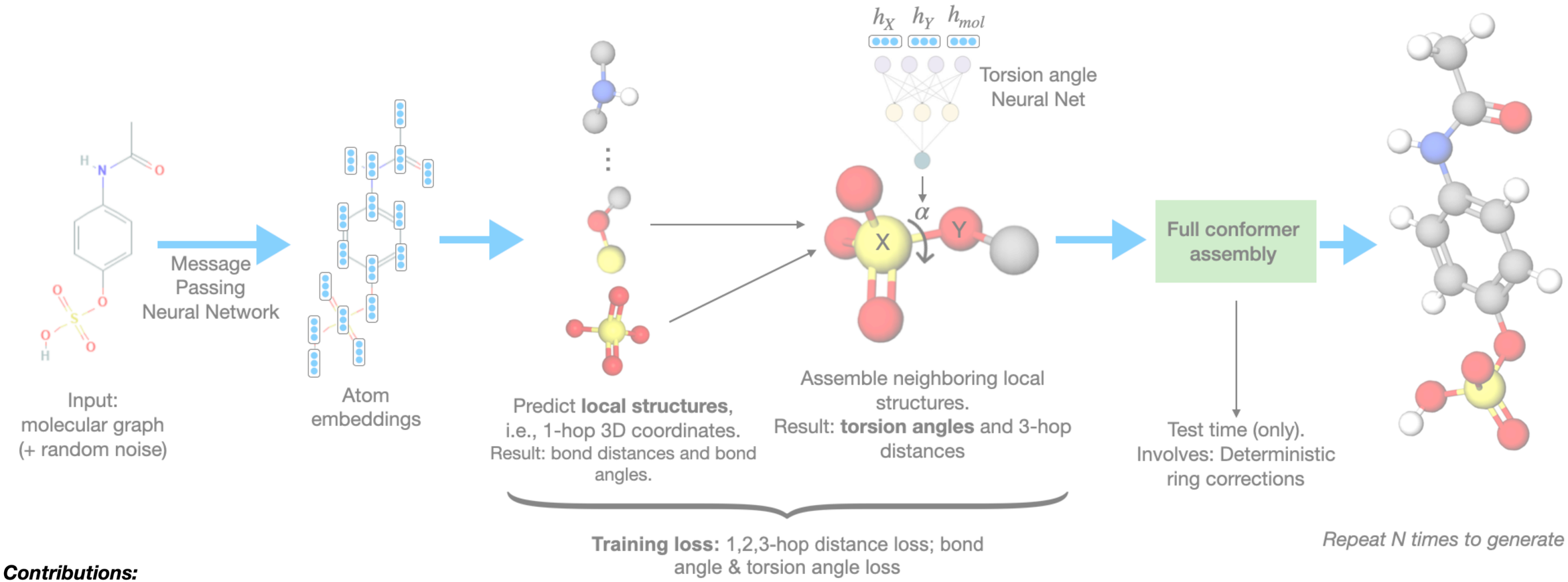
GeoMol - Overview



Contributions:

- Explicit prediction of bond distances, bond angles and torsion angles.

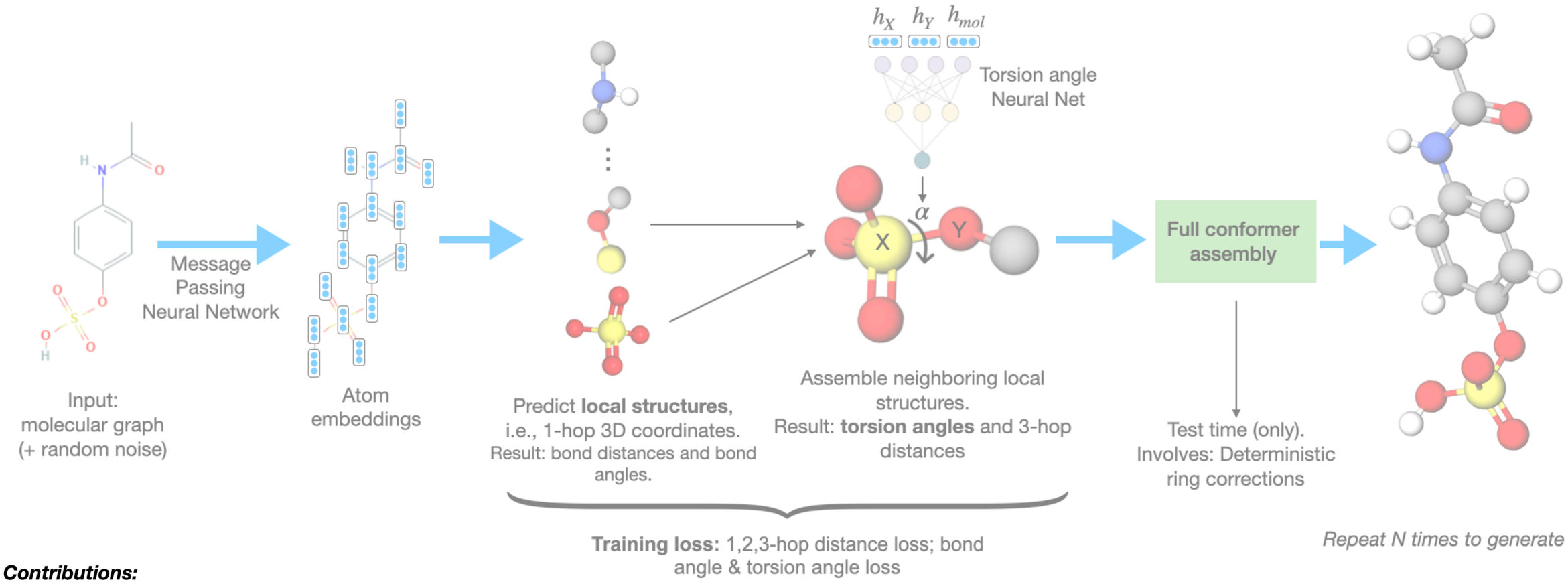
GeoMol - Overview



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

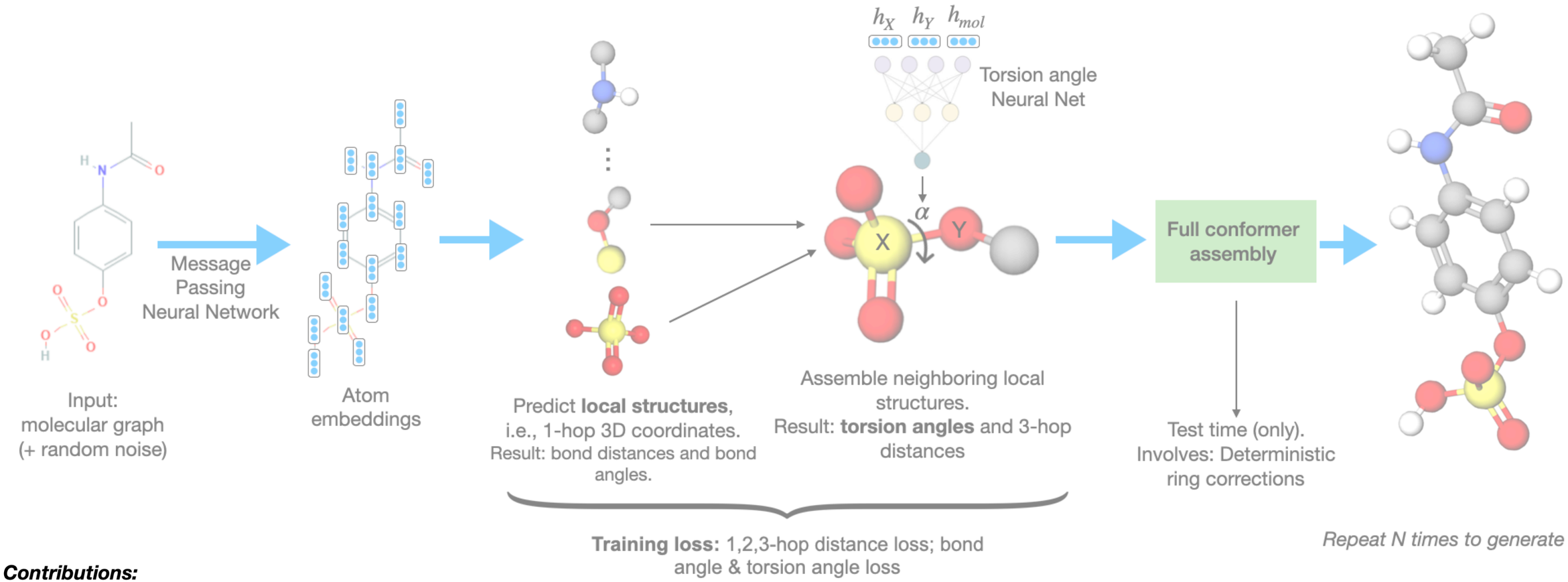
GeoMol - Overview



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 (to any global rotation/translation)

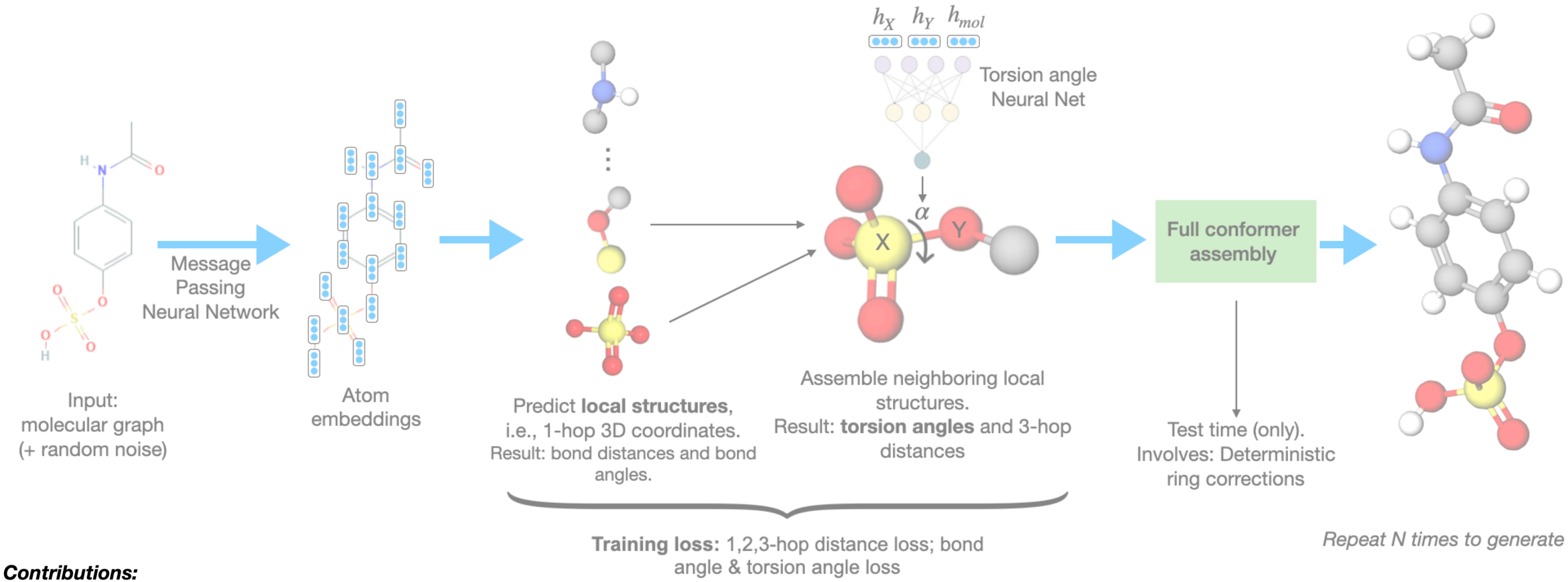
GeoMol - Overview



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 (to any global rotation/translation)
- Tetrahedral chiral centers are predicted exactly ; no iterative optimization necessary as with traditional distance geometry approaches

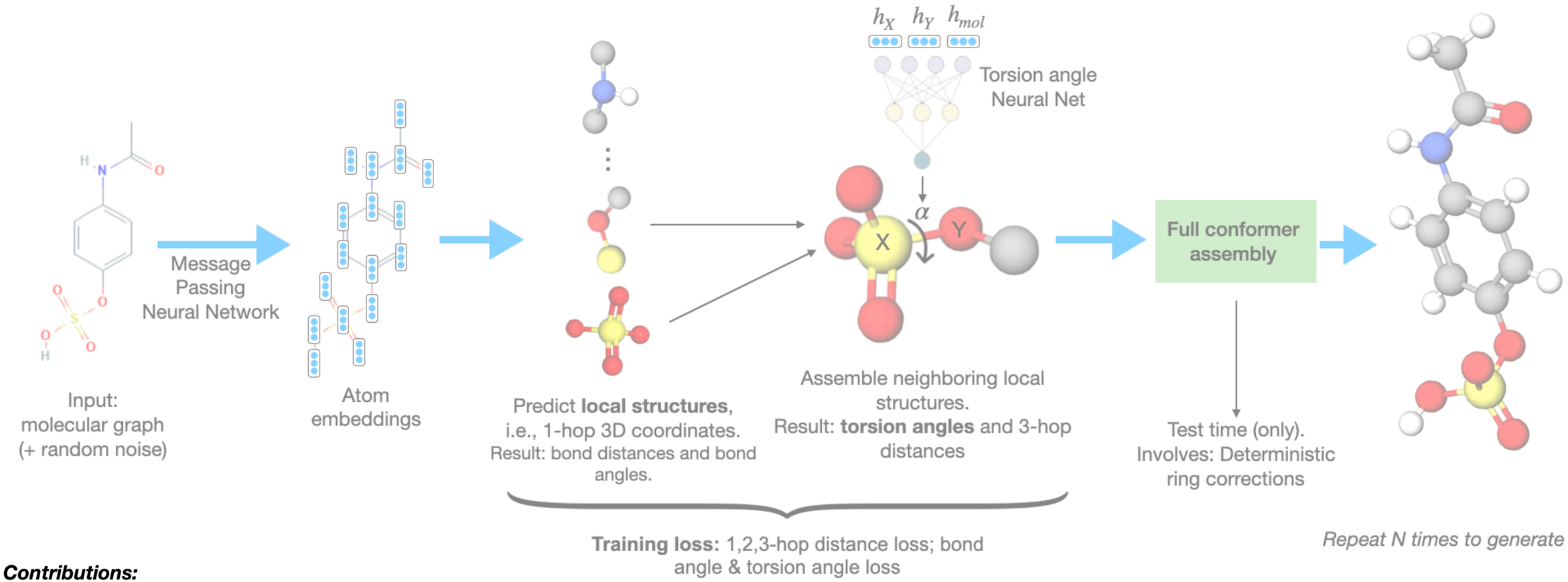
GeoMol - Overview



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 (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

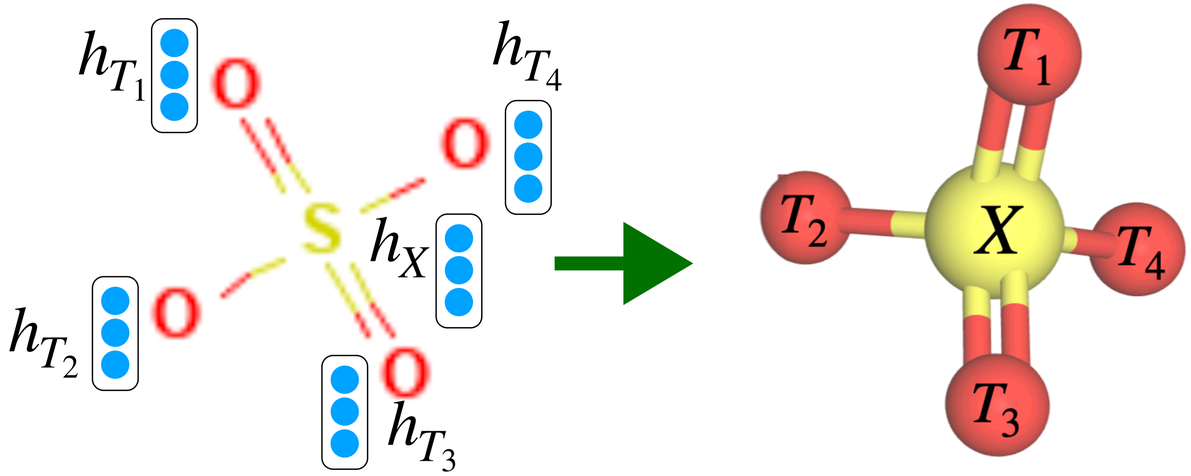
GeoMol - Overview



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 (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
- Conformers generated directly by the neural network, without iterative optimization such as Langevin dynamics

Local Structure (LS) Prediction

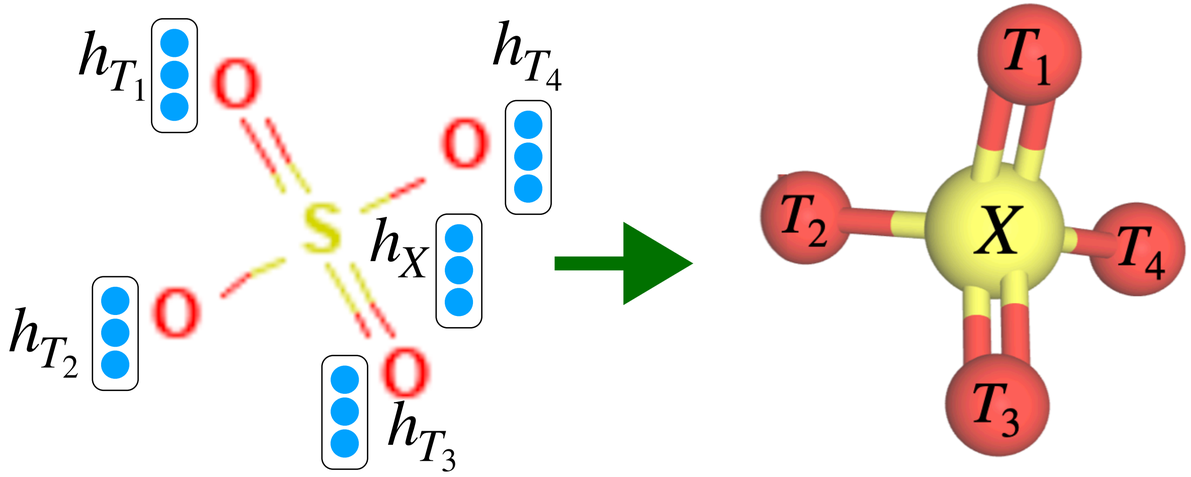


For each non-terminal atom X, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming X is placed in $\mathbf{0}$:

$$f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3\times n}$$

Result: bond distances and bond angles.

Local Structure (LS) Prediction



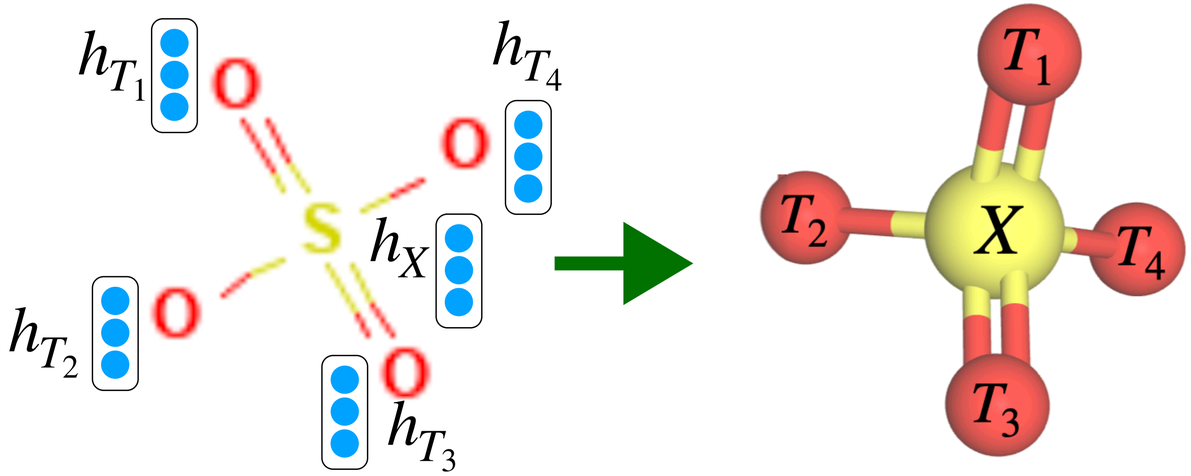
For each non-terminal atom X, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming X is placed in $\mathbf{0}$:

$$f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3\times n}$$

Challenges:

- 1) Equivariant prediction with respect to any permutation of neighbors: $f(\mathbf{h}_{T_{\pi(1)}}, \ldots, \mathbf{h}_{T_{\pi(n)}}; \mathbf{h}_X) = (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(n)}), \forall \pi \in S_n$

Local Structure (LS) Prediction



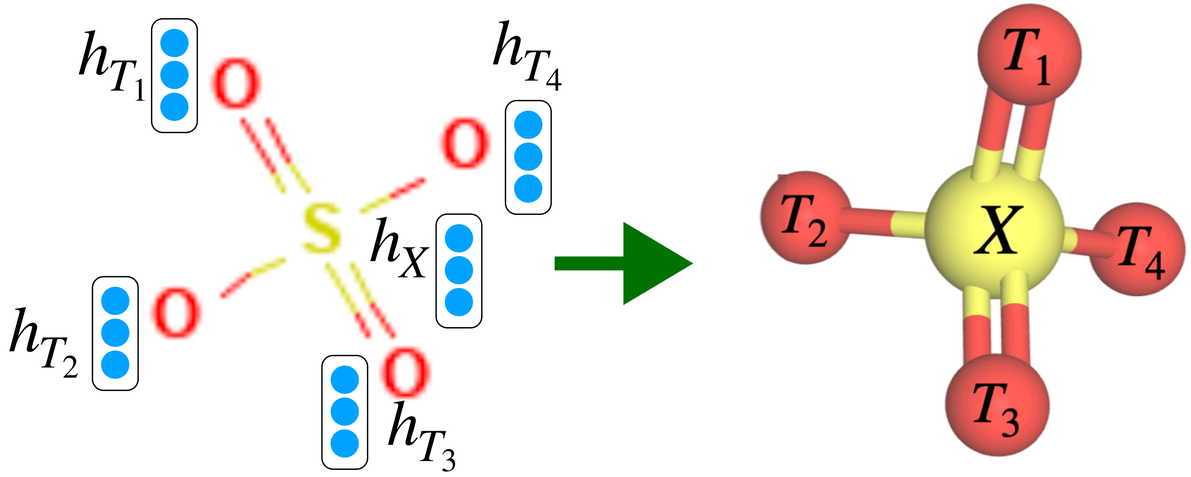
For each non-terminal atom X, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming X is placed in $\mathbf{0}$:

$$f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3\times n}$$

Challenges:

- 1) Equivariant prediction with respect to any permutation of neighbors: $f(\mathbf{h}_{T_{\pi(1)}}, \ldots, \mathbf{h}_{T_{\pi(n)}}; \mathbf{h}_X) = (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(n)}), \forall \pi \in S_n$
- 2) Bond distances should match symmetrically,
    - 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



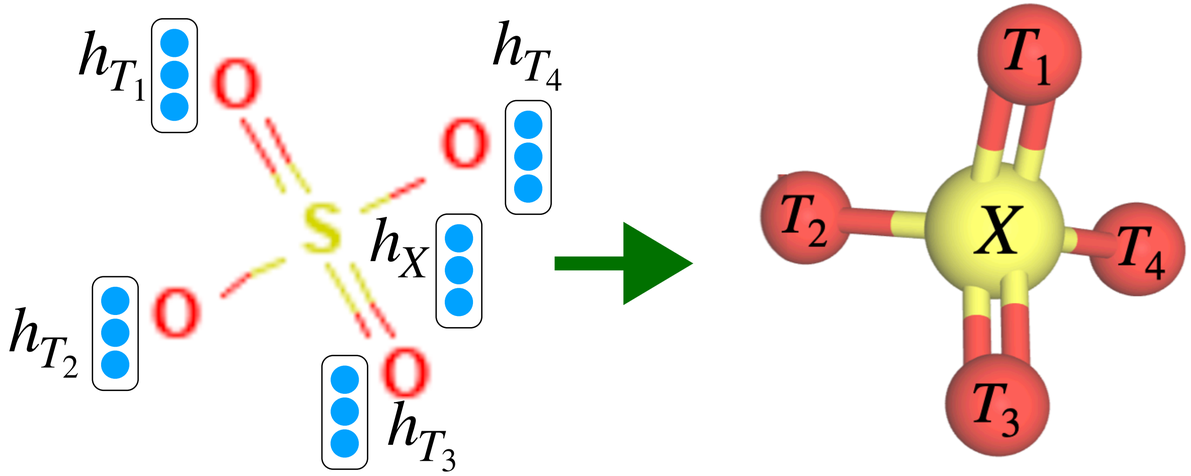
For each non-terminal atom X, predict the relative 3D coordinates of all its 1-hop neighbors (atom local 3D environment) assuming X is placed in $\mathbf{0}$:

$$f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3\times n}$$

Challenges:

- 1) Equivariant prediction with respect to any permutation of neighbors: $f(\mathbf{h}_{T_{\pi(1)}}, \ldots, \mathbf{h}_{T_{\pi(n)}}; \mathbf{h}_X) = (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(n)}), \forall \pi \in S_n$
- 2) Bond distances should match symmetrically,
  - i.e. XT_1 should have the same length predicted from the LS of X or from the LS of T_1
- **Solution**: a special symmetric transformer that separates distance prediction from direction prediction (see paper)

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 $\mathbf{0}$:

$$f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3\times n}$$

Challenges:

- 1) Equivariant prediction with respect to any permutation of neighbors: $f(\mathbf{h}_{T_{\pi(1)}}, \ldots, \mathbf{h}_{T_{\pi(n)}}; \mathbf{h}_X) = (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(n)}), \forall \pi \in S_n$
- 2) Bond distances should match symmetrically,
    - 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*

Tackling Chirality

- Chiral information:
  - Bond annotations to describe different molecules with same molecular graph, but different 3D structures (and, thus, different chemical behavior)
  - Differentiates mirroring structures
- Bond annotations are not fixed, i.e. multiple equivalent annotations

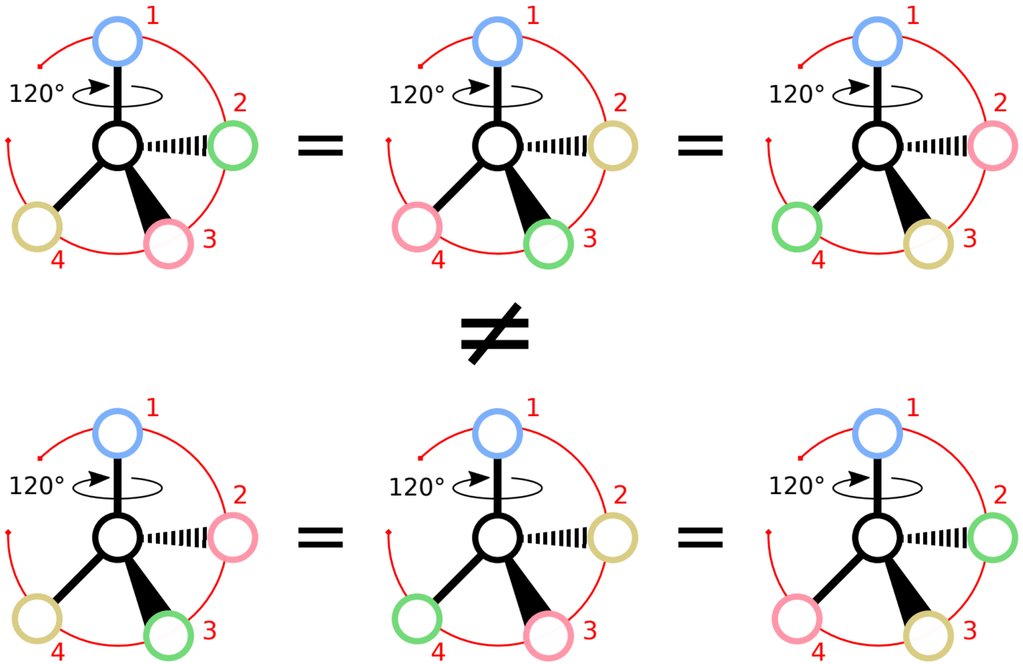
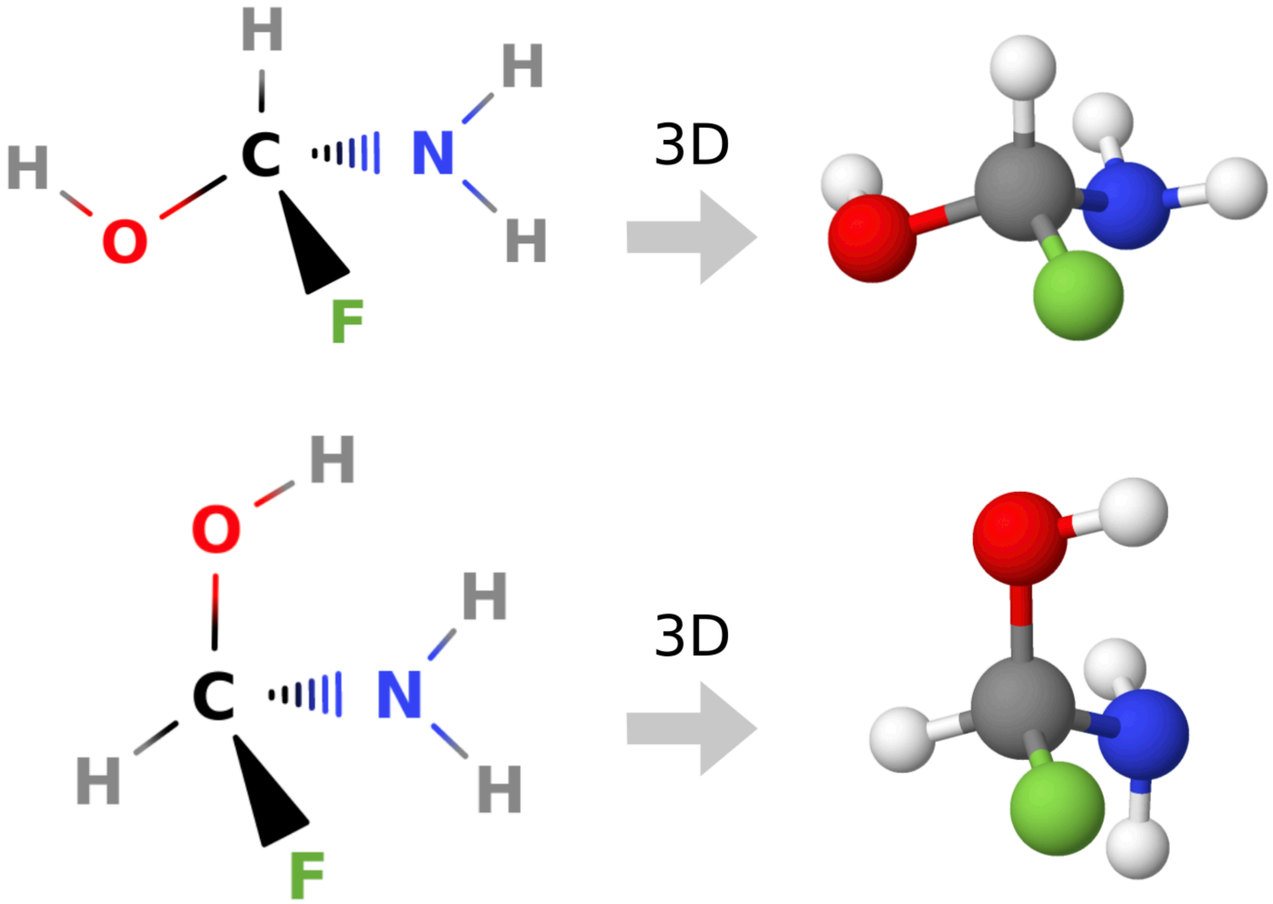


Fig source: “Message Passing Networks for Molecules with Tetrahedral Chirality”, L Pattanaik, OE Ganea et al, 2020

Tackling Chirality

- MPNNs cannot distinguish chirality …
  - … unless order of graph neighbors is explicitly used



Tackling Chirality Exactly

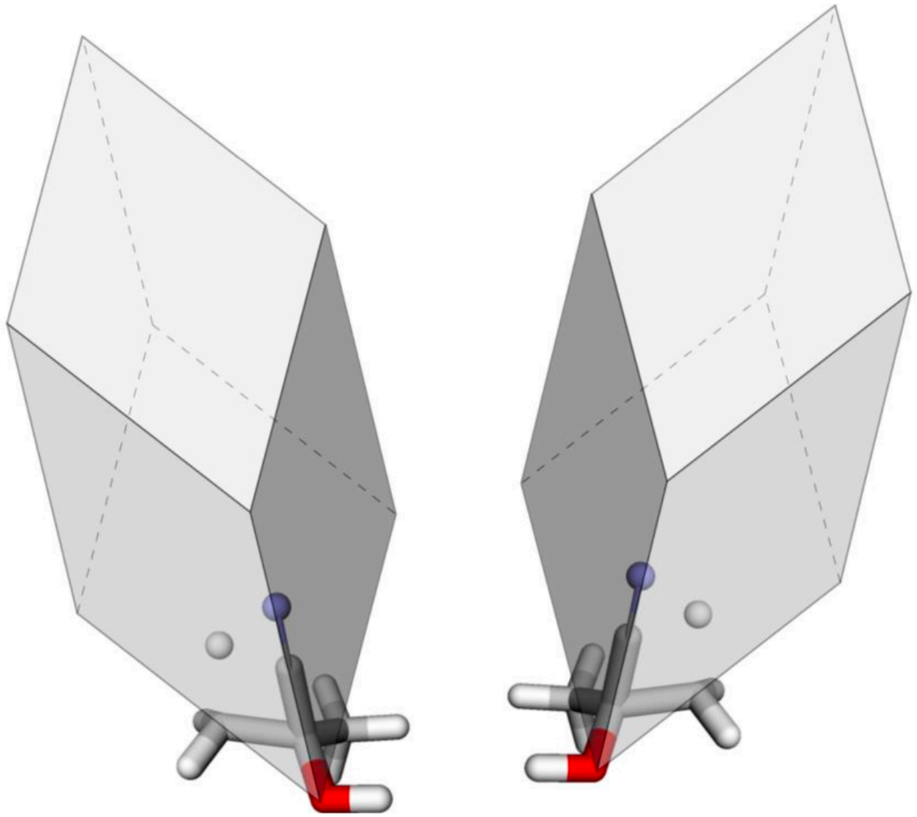
- Given a chiral center (or any center with 4 neighbors), we can compute the oriented volume.

$$OV(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) \stackrel{\text{def}}{=} sign\left(\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix}\right)$$

- 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.

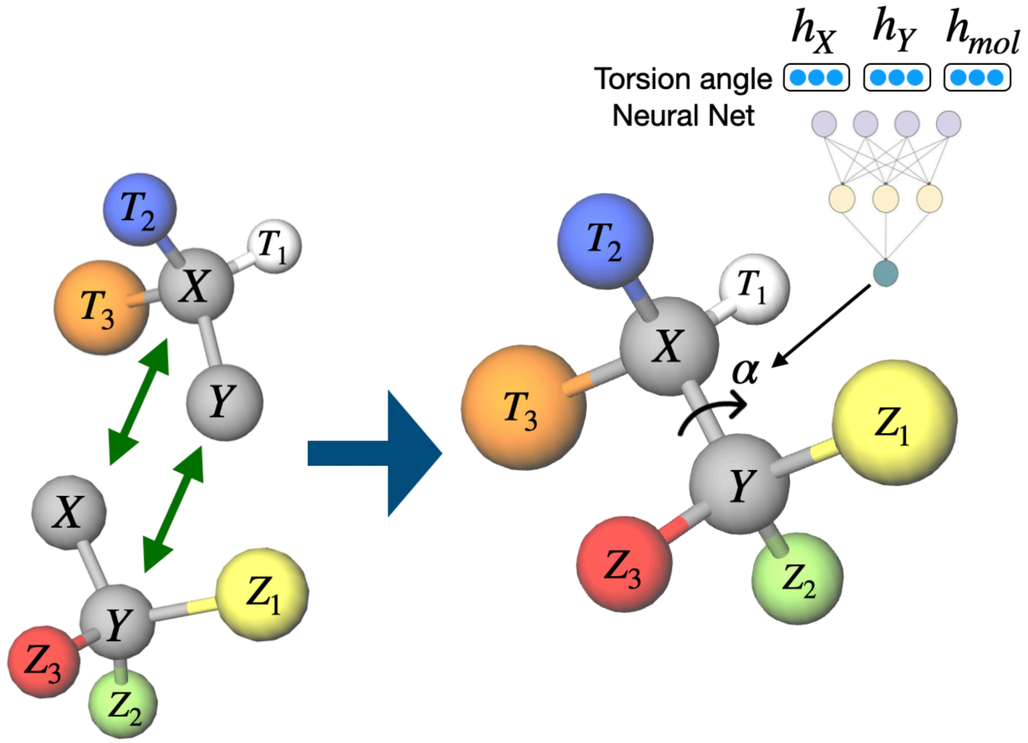
- No iterative optimization is needed.



$OV(C_S) = -1$ $OV(C_R) = +1$

Assembling Two Neighboring LS via Torsion Angle Prediction

Assemble every two local neighboring structures.

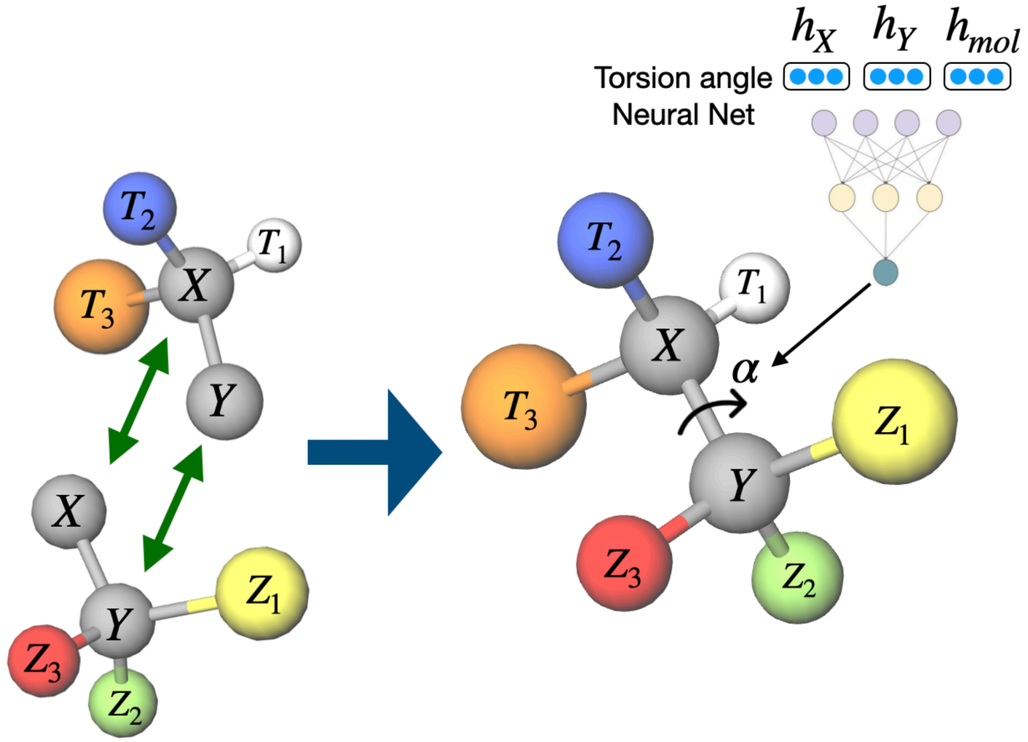


Assembling Two Neighboring LS via Torsion Angle Prediction

Assemble every two local neighboring structures.

Challenges:

- 1) Parameterize a single canonical torsion angle per rotatable bond (no over-parameterization)
    - All dihedral angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion



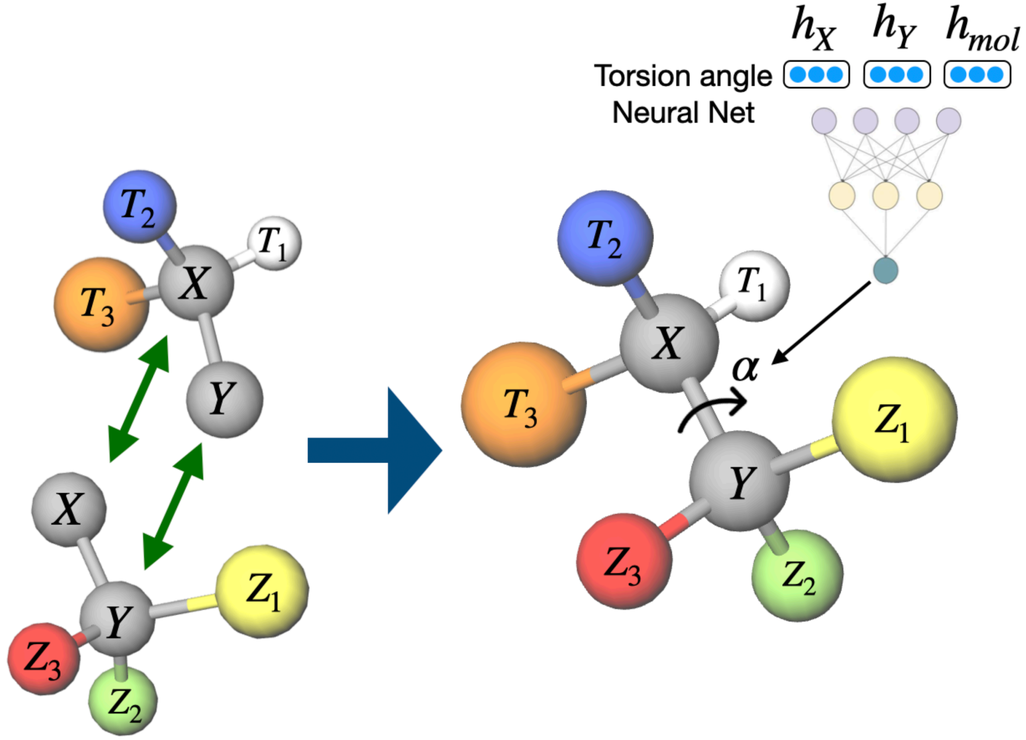
Assembling Two Neighboring LS via Torsion Angle Prediction

Assemble every two local neighboring structures.

Challenges:

- 1) Parameterize a single canonical torsion angle per rotatable bond (no over-parameterization)
    - All dihedral angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion

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



Assembling Two Neighboring LS via Torsion Angle Prediction

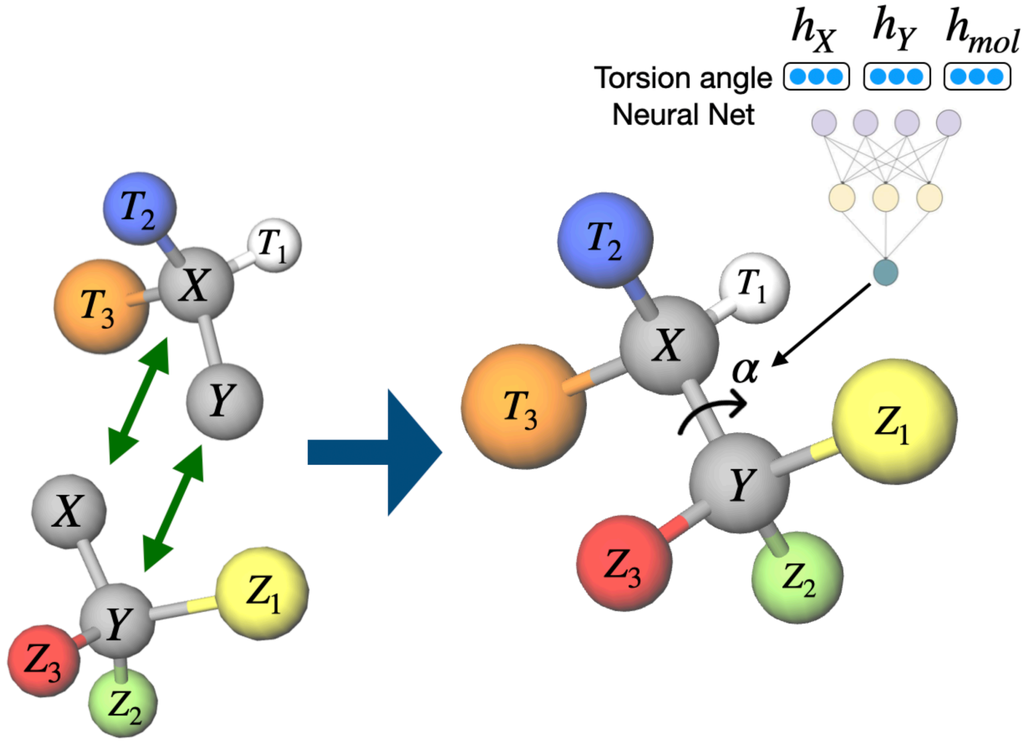
Assemble every two local neighboring structures.

Challenges:

- 1) Parameterize a single canonical torsion angle per rotatable bond (no over-parameterization)
    - All dihedral angles $\angle(XYT_i, XYZ_j)$ are coupled via a single canonical torsion

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

- **Solution** - novel Torsion Angle Neural Network - see our paper



Optimal Transport Loss

- First setting: one predicted conformer C and one ground truth conformer C^*: loss $\mathscr{L}(C, C^*)$
    - Matches 1,2,3-hop distances, bond and torsion angles.

Optimal Transport Loss

- First setting: one predicted conformer C and one ground truth conformer C^*: loss $\mathscr{L}(C, C^*)$
  - Matches 1,2,3-hop distances, bond and torsion angles.

- 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 $\mathscr{L}(C, C^*)$?
  - How to avoid adversarial training (impractical, hard to train, expensive) ?
  - How to generate diverse conformers (to cover all modes of the true distribution) ?

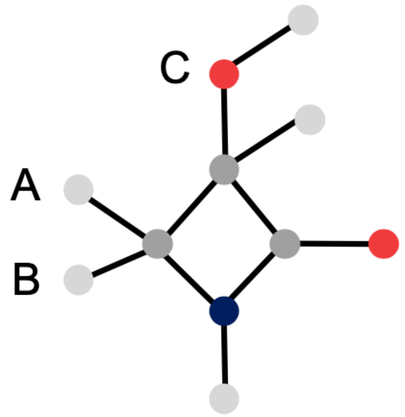
Optimal Transport Loss

- First setting: one predicted conformer C and one ground truth conformer C^*: loss $\mathscr{L}(C, C^*)$
  - Matches 1,2,3-hop distances, bond and torsion angles.

- 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 $\mathscr{L}(C, C^*)$?
  - How to avoid adversarial training (impractical, hard to train, expensive) ?
  - How to generate diverse conformers (to cover all modes of the true distribution) ?
  - **Solution**: Wasserstein loss: $\mathscr{L} = \mathscr{W}_{\mathscr{L}(\cdot,\cdot)}(\{C_j\}_j, \{C_i^*\}_i) = \inf_{\substack{\mathbf{T} \in \mathbb{R}^{m \times n} \\ \mathbf{T}\ doubly\ stochastic}} \sum_{i=1}^{n} \sum_{j=1}^{m} T_{ij} \mathscr{L}(C_j, C_i^*)$

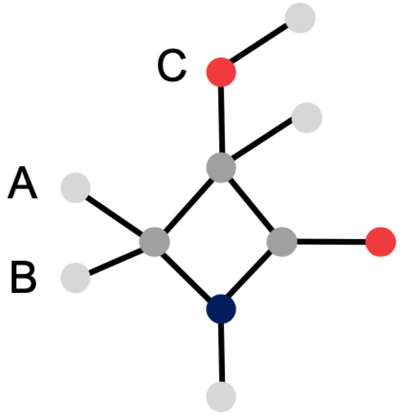
Dealing with Symmetry

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



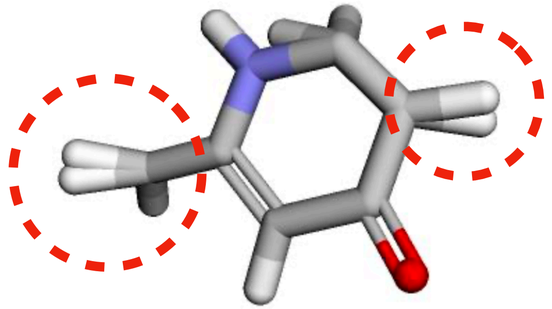
Dealing with Symmetry

- 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:

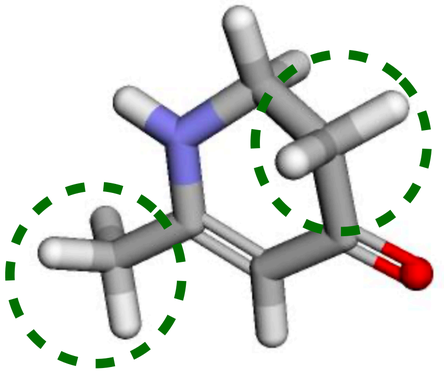


$$\mathscr{L}^{perm}(C, C^*) := \min\left(\mathscr{L}(C_{A,B}, C^*_{A,B}), \mathscr{L}(C_{B,A}, C^*_{A,B})\right)$$

without permutation-dependent loss:



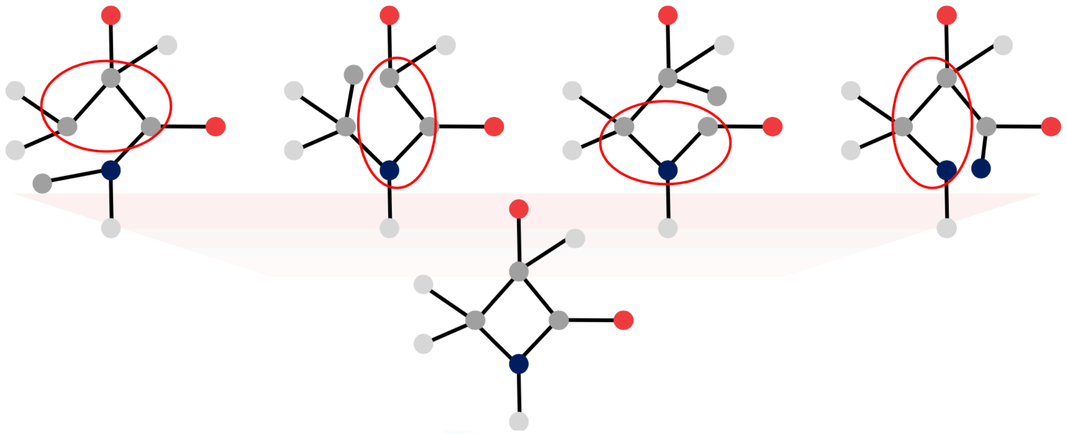
with permutation-dependent loss:

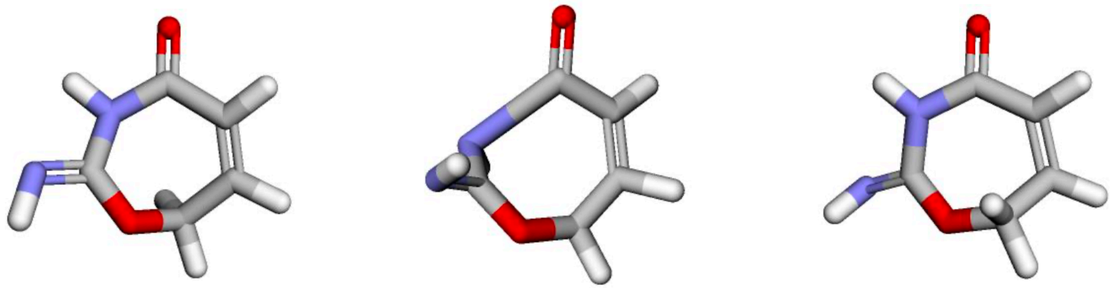


Assemble Full Conformer at Test Time

- We can assemble any tree-like molecule using predicted local structures and torsion angles.

- We correct rings by averaging over all spanning trees and using Kabsch superimposition algorithm.





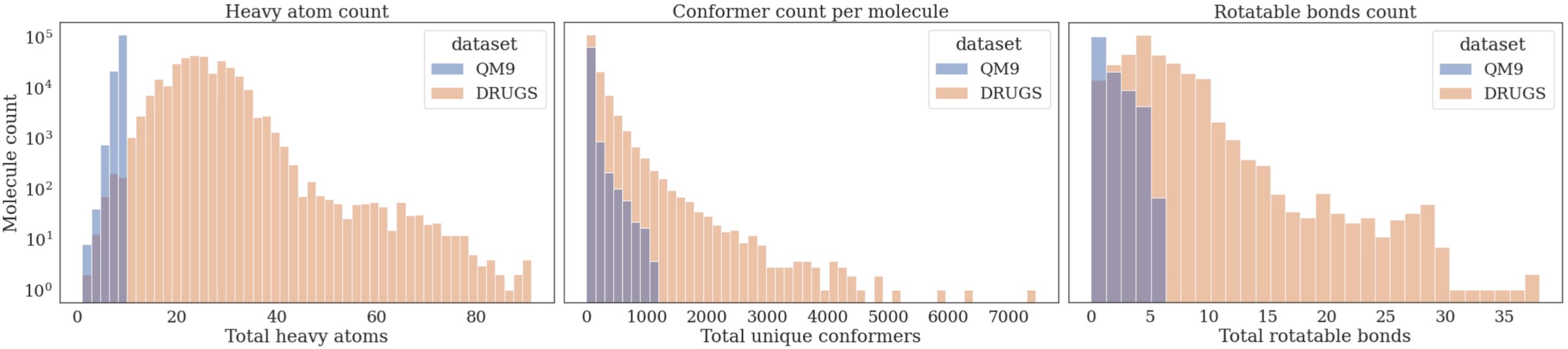
ground truth

model without ring correction

model with ring correction

Datasets

- **Datasets:**
  - GEOM-QM9 conformers: small molecules
  - GEOM-DRUGS (AG, 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):

- Comparison of two conformer distributions:
  - **Coverage** ↑: percentage of “correctly” generated conformers from ground truth set.

$$\text{COV - R (Recall)} \stackrel{\text{def}}{=} \frac{1}{L} |\{l \in [1..L] : \exists k \in [1..K], RMSD(\mathcal{C}_k, \mathcal{C}_l^*) < \delta\}|$$

  - **Average Minimum RMSD** ↓: for each generated conformer, compute RMSD to the closest ground truth. Average over all.

$$\text{AMR - R (Recall)} \stackrel{\text{def}}{=} \frac{1}{L} \sum_{l \in [1..L]} \min_{k \in [1..K]} RMSD(\mathcal{C}_k, \mathcal{C}_l^*)$$

  - **COV - P** (Precision) and **AMR - P** (Precision) defined similarly

Results (no FF fine-tuning)

- For each molecule, we ask models to generate 2x as many conformers as in the ground truth
- Test set: 1000 molecules

Table 1: Results on the **GEOM-DRUGS** dataset.

	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
Models	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GraphDG *(ML)*	10.37	0.00	1.950	1.933	3.98	0.00	2.420	2.420
CGCF *(ML)*	54.35	56.74	1.248	1.224	24.48	15.00	1.837	1.829
RDKit/ETKDG	68.78	76.04	1.042	0.982	71.06	88.24	1.036	0.943
OMEGA *(C)*	81.64	97.25	0.851	**0.771**	77.18	**96.15**	0.951	**0.854**
GEOMOL ($s = 9.5$)	**86.07**	**98.06**	**0.846**	0.820	71.78	83.77	1.039	0.982
GEOMOL ($s = 5$)	82.43	95.10	0.862	0.837	**78.52**	94.40	**0.933**	**0.856**

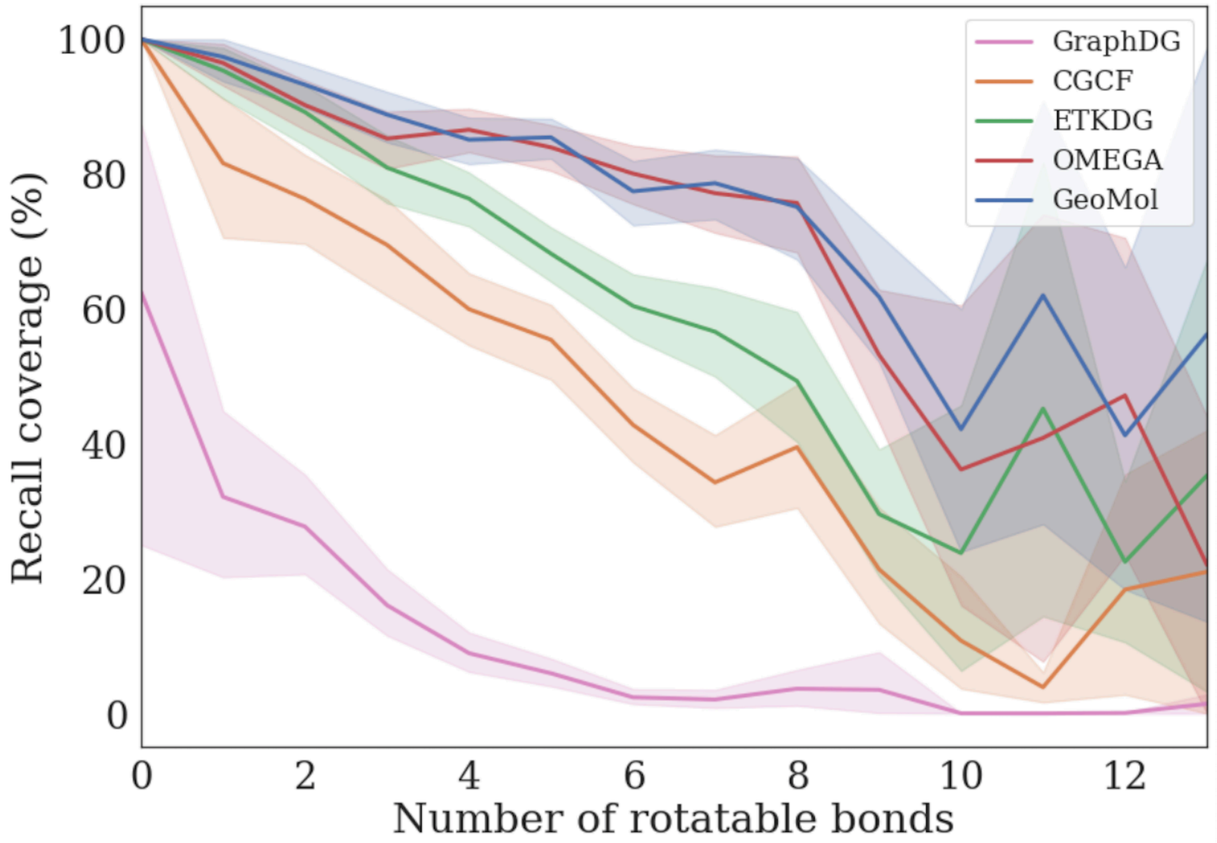
Table 2: Results on the **GEOM-QM9** dataset.

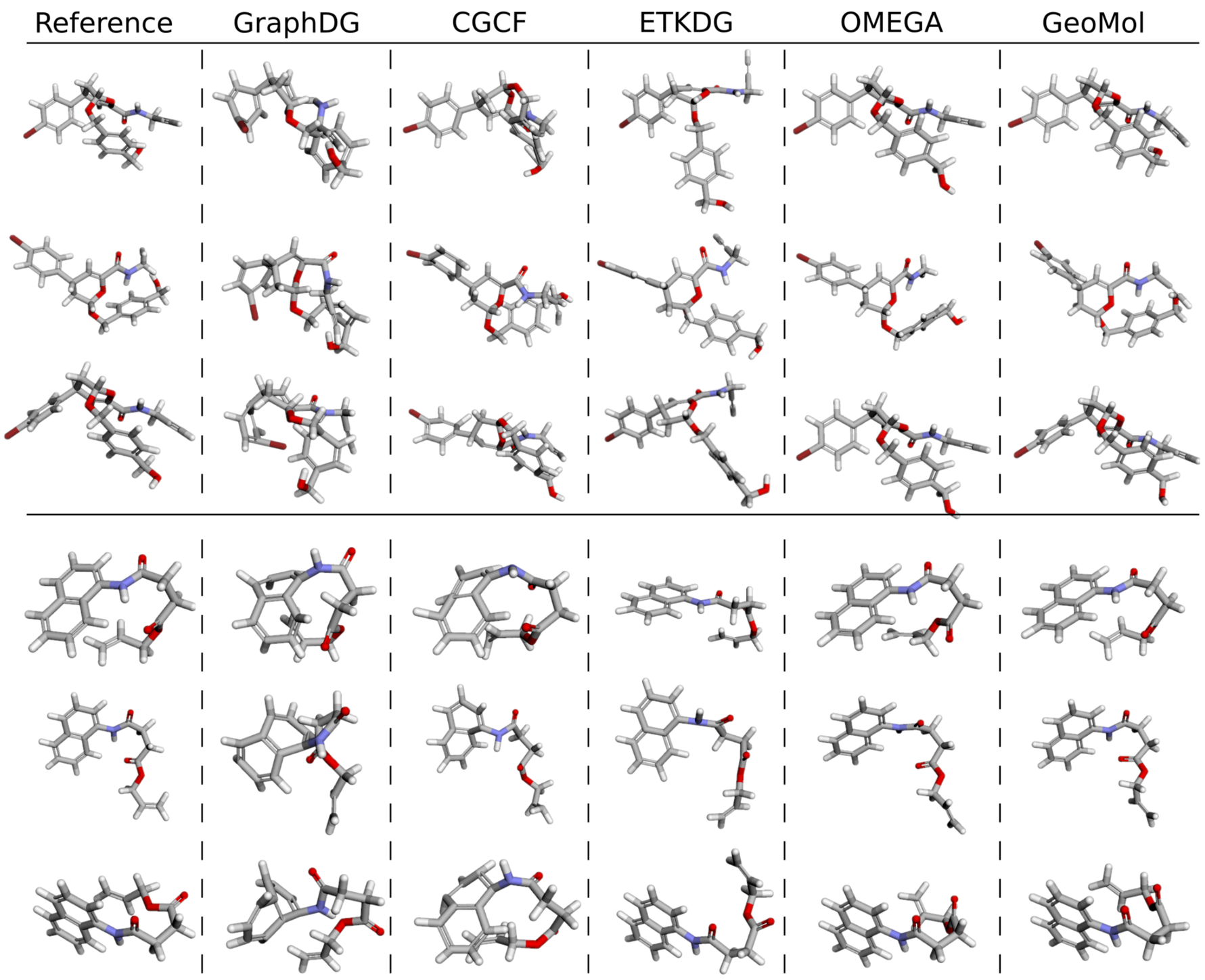
	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
Models	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GraphDG *(ML)*	74.66	100.00	0.373	0.337	63.03	77.60	0.450	0.404
CGCF *(ML)*	69.47	96.15	0.425	0.374	38.20	33.33	0.711	0.695
RDKit/ETKDG	85.13	**100.00**	0.235	0.199	**86.80**	**100.00**	0.232	0.205
OMEGA *(C)*	85.51	**100.00**	**0.177**	**0.126**	82.86	**100.00**	**0.224**	**0.186**
GEOMOL ($s = 5$)	**91.52**	**100.00**	0.225	0.193	**86.71**	**100.00**	0.270	0.241

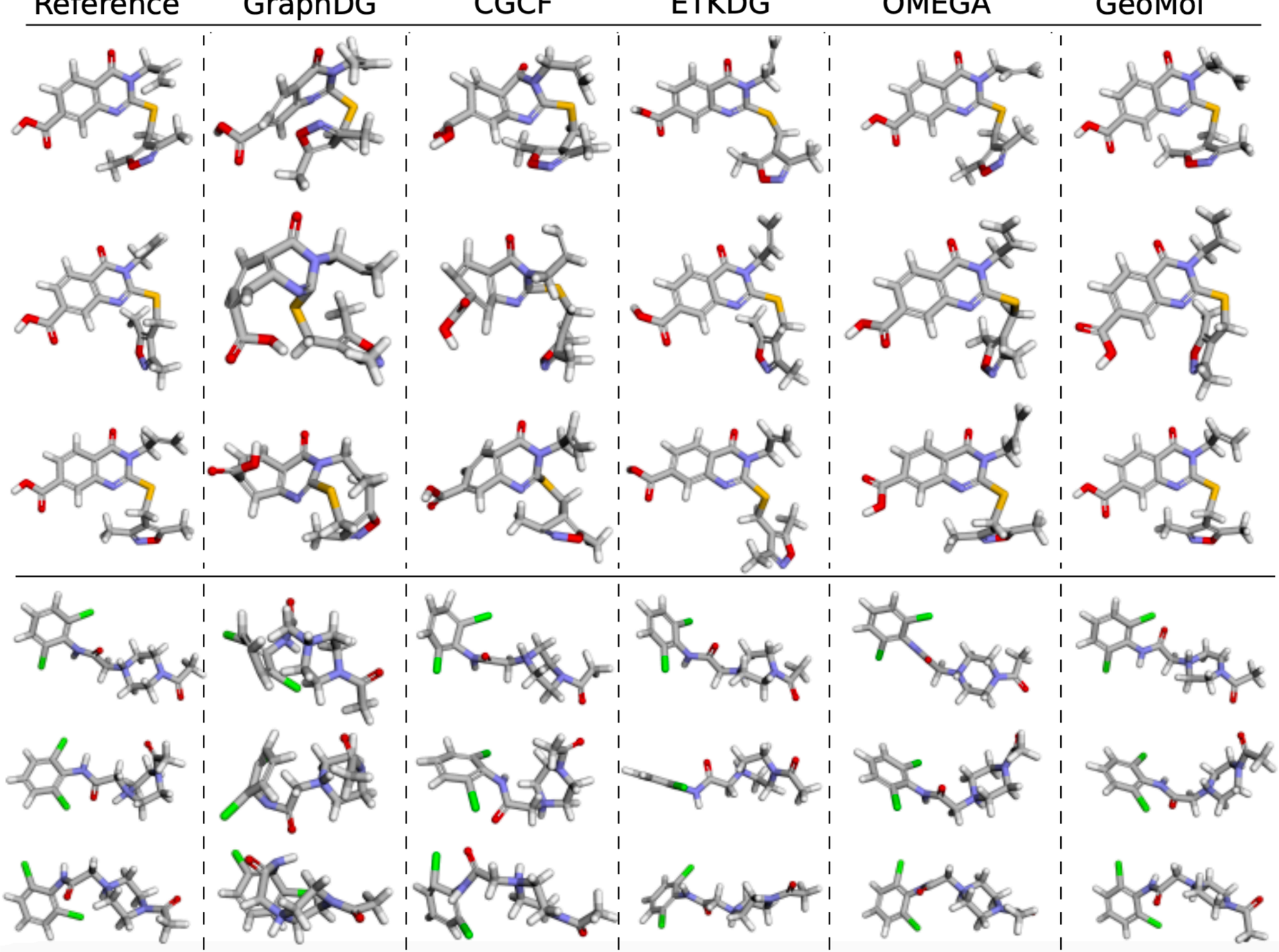
GraphDG: Simm, Gregor NC, and José Miguel Hernández-Lobato. "A generative model for molecular distance geometry.", ICML 2020

CGCF: Learning neural generative dynamics for molecular conformer generation. ICLR 2021

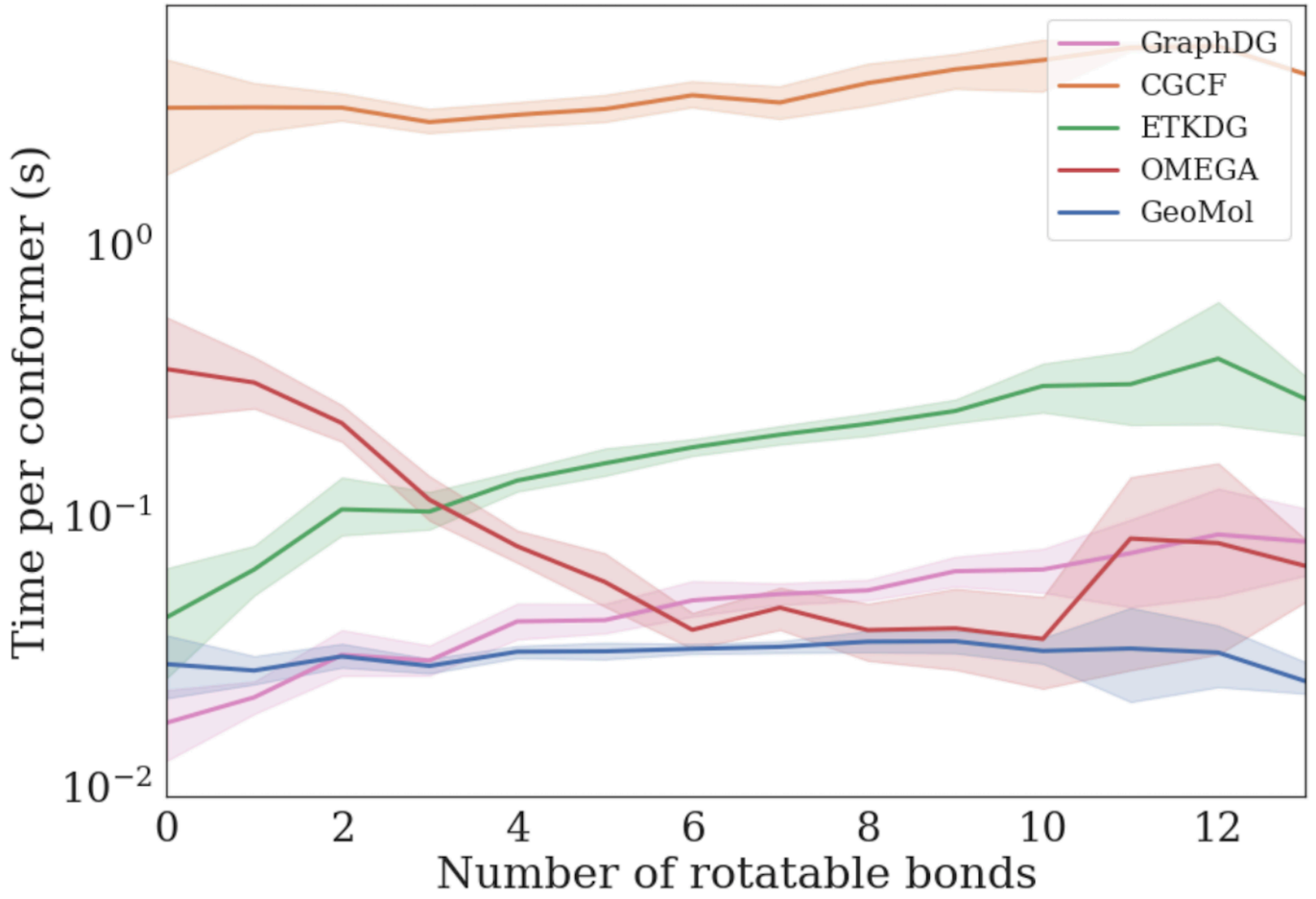
Num of rotatable bonds vs coverage




Reference
GraphDG
CGCF
ETKDG
OMEGA
GeoMol

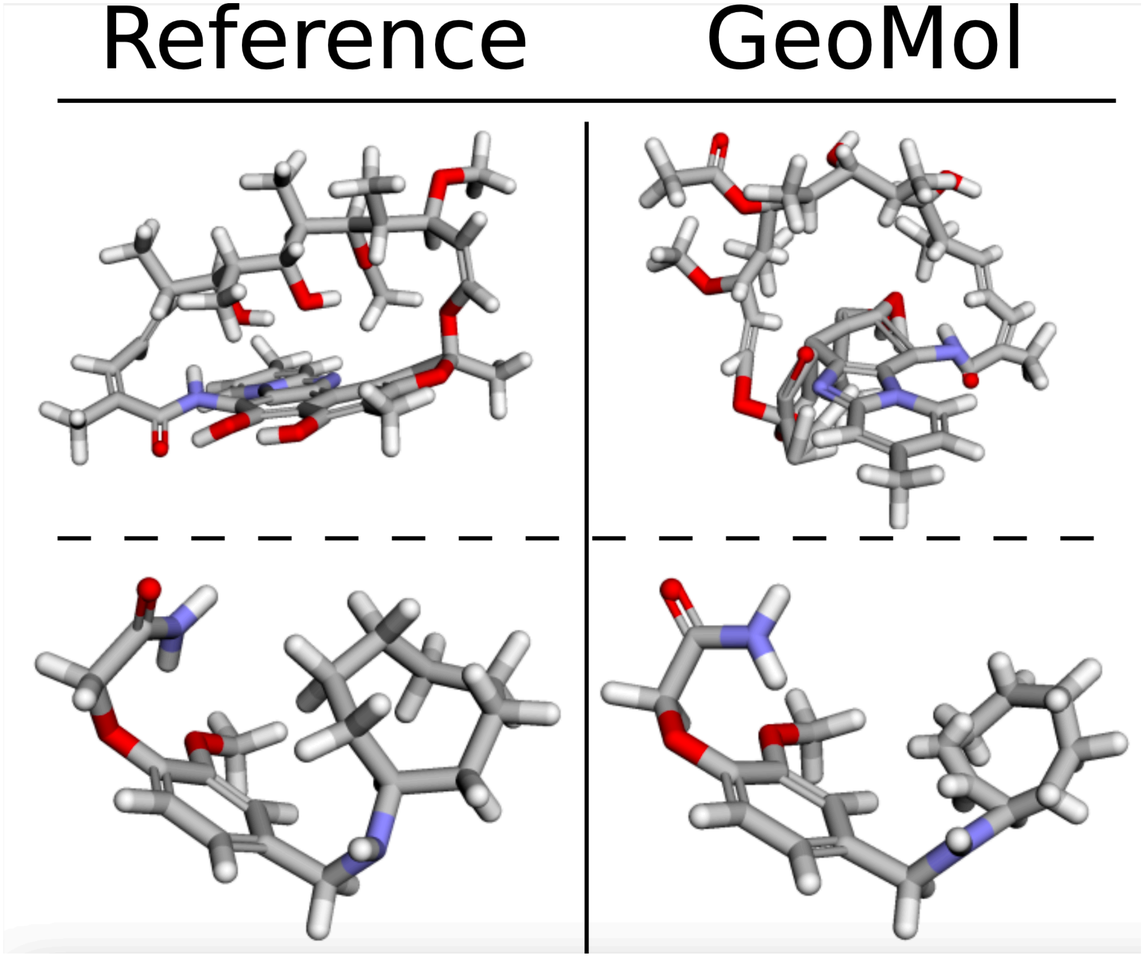

Reference
GraphDG
CGCF
ETKDG
OMEGA
GeoMol

Conformer Generation Time



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!