GEOMOL: Torsional Geometric Generation of Molecular 3D Conformer Ensembles

Octavian-Eugen Ganea [*,‡] **Lagnajit Pattanaik** [*,†]

Connor W. Coley [†] **Regina Barzilay** [‡] **Klavs F. Jensen** [†]

William H. Green [†] **Tommi S. Jaakkola** [‡]

Abstract

Prediction of a molecule's 3D conformer ensemble from the molecular graph holds a key role in areas of cheminformatics and drug discovery. Existing generative models have several drawbacks including lack of modeling important molecular geometry elements (e.g. torsion angles), separate optimization stages prone to error accumulation, and the need for structure fine-tuning based on approximate classical force-fields or computationally expensive methods such as metadynamics with approximate quantum mechanics calculations at each geometry. We propose GEOMOL–an end-to-end, non-autoregressive and SE(3)-invariant machine learning approach to generate distributions of low-energy molecular 3D conformers. Leveraging the power of message passing neural networks (MPNNs) to capture local and global graph information, we predict local atomic 3D structures and torsion angles, avoiding unnecessary over-parameterization of the geometric degrees of freedom (e.g. one angle per non-terminal bond). Such local predictions suffice both for the training loss computation, as well as for the full deterministic conformer assembly (at test time). We devise a non-adversarial optimal transport based loss function to promote diverse conformer generation. GEOMOL predominantly outperforms popular open-source, commercial, or state-of-the-art machine learning (ML) models, while achieving significant speed-ups. We expect such differentiable 3D structure generators to significantly impact molecular modeling and related applications. [4]

1 Overview

Problem & importance. We tackle the problem of molecular conformer generation (MCG), i.e. predicting the ensemble of low-energy 3D conformations of a small molecule solely based on the molecular graph (fig. 1). A single conformation is represented by the list of 3D coordinates for each atom in the respective molecule. In this work, we assume that the low-energy states are implicitly defined by the given dataset, i.e. our training data consist of molecular graphs and corresponding sets of energetically favorable 3D conformations. Low-energy structures are the most stable configurations and, thus, expected to be observed most often experimentally.

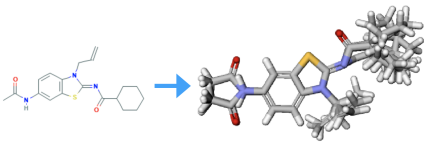

Figure 1: We explore the task of low-energy 3D conformer ensemble generation from the input molecular graph. This example molecule has both rigid (rings) and more flexible parts. Conformers are shown aligned and juxtaposed.

[*]Equal first authorship and contribution. Correspondence to {oct,lagnajit}@mit.edu .

[†]Department of Chemical Engineering, MIT, Cambridge, MA 02139

[‡]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139

[4]Code is available at https://github.com/PattanaikL/GeoMol.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Dealing with molecules in their natural 3D structure is of great importance in areas such as cheminformatics or computational drug discovery because conformations determine biological, chemical, and physical properties [Guimaraes et al., 2012, Schütt et al., 2018, Klicpera et al., 2019, Axelrod and Gomez-Bombarelli, 2020b, Schütt et al., 2021, Liu et al., 2021] such as charge distribution, potential energy, docking poses [McGann, 2011], shape similarity [Kumar and Zhang, 2018], pharmacophore searching [Schwab, 2010], or descriptors for 3/4D QSAR [Verma et al., 2010]. For instance, in drug design it is crucial to understand how a molecule binds to a specific target protein; this process heavily depends on the 3D structures of the two components, both in terms of geometric (shape matching) and chemical (hydrophobic/hydrophilic) interactions [Gainza et al., 2020, Sverrisson et al., 2020].

Motivation & challenges of existing methods. The main challenge in MCG comes from the enormous size of the 3D structure space consisting of bond lengths, bond angles, and torsion angles. It is known that the molecular graph imposes specific constraints on possible 3D conformations, e.g. bond length ranges depend on the respective bond types, while chiral centers dictate local spatial arrangement. However, the space of possible conformations grows exponentially with the graph size and number of rotatable bonds, thus hindering exhaustive brute force exploration even for relatively small molecules. Additionally, the number of plausibly-stable low-energy states is unknown a priori and can vary between one and several thousand conformations for a single molecule [Chan et al., 2021]. Nevertheless, various facets of the curse of dimensionality have been favorably tackled by ML models in different contexts, and our goal is to build on the recent ML efforts for MCG [Mansimov et al., 2019, Simm and Hernandez-Lobato, 2020, Lemm et al., 2021, Xu et al., 2021].

Molecular conformations can be determined experimentally, but existing techniques are very expensive. As a consequence, predictive computational models have been developed over the past few decades, traditionally being categorized as either *stochastic* or *systematic* (rule-based) methods [Hawkins, 2017]. Stochastic approaches have traditionally been based on molecular dynamics (MD) or Markov chain Monte Carlo (MCMC) techniques, potentially combined with genetic algorithms (GAs). They can do extensive explorations of the energy landscape and accurately sample equilibrium structures, but quickly become prohibitively slow for larger molecules [Shim and MacKerell Jr, 2011, Ballard et al., 2015, De Vivo et al., 2016, Hawkins, 2017], e.g. they require several CPU minutes for a single drug-like molecule. Moreover, stochastic methods have difficulties sampling diverse and representative conformers, prioritizing quantity over quality. On the other hand, rule-based systematic methods achieve state-of-the-art in commercial software [Friedrich et al., 2017] with OMEGA [Hawkins et al., 2010, Hawkins and Nicholls, 2012] being a popular example. They usually process a single drug-like molecule under a second. They address the aforementioned challenges of stochastic methods by relying on carefully curated torsion templates (torsion rules), rule-based generators, and knowledge bases of rigid 3D fragments, which are assembled together and combined with subsequent stability score ranking. However, torsion angles are mostly varied independently (based on their fragments), without explicitly capturing their global interactions, which results in difficulties for larger and more flexible molecules. Furthermore, the curated fragments and rules are inadequate for more challenging inputs (e.g. transition states or open-shell molecules).

Both types of methods can be combined with Distance Geometry (DG) techniques to generate the initial 3D conformation. First, the 3D atom distance matrix is generated based on a set of distance constraints or from a specialized model. Subsequently, the corresponding 3D atom coordinates are learned to approximately match these predicted distances [Havel et al., 1983b,a, Crippen et al., 1988, Havel, 1998, Lagorce et al., 2009, Riniker and Landrum, 2015]. Indeed, modern stochastic algorithms are entirely based on DG methods [Riniker and Landrum, 2015]. The inductive bias of rotational and translational invariance is guaranteed for DG, thus being appealing for ML models [Simm and Hernandez-Lobato, 2020, Xu et al., 2021, Pattanaik et al., 2020b]. However, several drawbacks weaken this important direction: i) the distance matrix is overparameterized compared to the actual number of degrees of freedom, ii) it is difficult to enforce 3D Euclidean distance constraints as well as geometric graph constraints (e.g. on torsion angles or rings [Riniker and Landrum, 2015]); iii) important aspects of molecular geometry are not explicitly modeled, e.g. torsion angles of rotatable bonds or chiral centers; iv) expensive force-field energy fine-tuning of the generated conformers is vital for a reasonable quality [Xu et al., 2021, Simm and Hernandez-Lobato, 2020]; iv) the resulting multi-stage pipeline is prone to error accumulation as opposed to an end-to-end model.

Previous methods often rely on a force field (FF) energy function minimization to fine-tune the conformers. These are hand-designed energy models which use parameters estimated from experi-

ment and/or computed from quantum mechanics (e.g. Universal Force Field [Rappé et al., 1992], Merck Molecular Force Field [Halgren, 1996]). However, FFs are crude approximations of the true molecular potential energy surface [Kanal et al., 2018], limited in the interactions they can capture in biomolecules due to their strong assumptions [Barman et al., 2015]. In addition, FF energy optimization is relatively slow and increases error accumulation in a multi-pipeline method.

Relation to protein folding. There has been impressive recent progress on modeling protein folding dynamics [Ingraham et al., 2018, AlQuraishi, 2019, Noé et al., 2019, Senior et al., 2020], where crystallized 3D structures are predicted solely from the amino-acid sequence using ML methods. However, molecules pose unique challenges, being highly branched graphs containing cycles, different types of bonds, and chirality information. This makes protein folding approaches not readily transferable to general molecular data.

Our key contributions & model in a nutshell. In this work, we investigate the question:

Can we design fast ML generative models of high quality, representative, diverse, and generalizable low-energy 3D conformational ensembles from molecular graphs?

To tackle this question, we propose GEOMOL (shown in fig. 2), exhibiting the following merits:

- It is end-to-end trainable, non-autoregressive, and does not rely on DG techniques (thus avoiding aforementioned drawbacks). More precisely, it outputs a minimal set of geometric quantities (i.e. angles and distances) sufficient for full deterministic reconstruction of the 3D conformer.
- It models conformers in an SE(3)-invariant (translation/rotation) manner by design. This desirable inductive bias was previously either achieved using multi-step DG methods [Simm and Hernandez-Lobato, 2020] or not captured at all [Mansimov et al., 2019].
- It explicitly models and predicts essential molecular geometry elements: torsion angles and local 3D structures (bond distances and bond angles adjacent to each atom). Together with the input molecular graph, these are used for k-hop distance computation at train time and full deterministic conformation assembly at test time. Crucially, we do not over-parameterize these predictions, i.e. a single torsion angle is computed per each non-terminal bond, irrespective of the number and permutation of the neighboring atoms at each end-point of the respective bond.
- The above geometric elements (torsion angles, local structures) are SE(3)-invariant (by definition or usage) and we jointly predict them using MPNNs [Gilmer et al., 2017] and self-attention networks. Thus, unlike [Mansimov et al., 2019], we are not affected by MPNNs' pitfalls that obstruct direct predictions of 3D atom coordinates from node embeddings, e.g. symmetric or locally isomorphic nodes would always have identical MPNN embeddings [Xu et al., 2019, Garg et al., 2020] and, as a consequence, would be inappropriately assigned identical 3D coordinates.
- To promote diverse conformer ensembles with good coverage, we devise a tailored generative loss that does not use slow or difficult-to-optimize adversarial training techniques. Using optimal transport, GEOMOL finds the best matching between generated and ground truth conformers based on their pairwise log-likelihood loss, requiring only minimization.
- It explicitly and deterministically distinguishes reflected structures (enantiomers) by solving tetrahedral chiral centers using oriented volumes and local chiral descriptors, bypassing the need for iterative optimization usually done in DG approaches.
- Empirically, we conduct experiments on two benchmarks: GEOM-QM9 (smaller molecules relevant to gas-phase chemistry) and GEOM-DRUGS (drug-like molecules) [Axelrod and Gomez-Bombarelli, 2020a]. Our method often outperforms previous ML and two popular open-source or commercial methods in different metrics. Moreover, we show competitive quality even without the frequently-used computationally-demanding fine-tuning FF strategies.
- GEOMOL processes drug-like molecules in seconds or less, being orders of magnitude faster than popular baselines (e.g. ETKDG/RDKit[Riniker and Landrum, 2015]), without sacrificing quality.

2 Method

Problem setup & notations. Our input is any molecular graph $G = (V, E)$ with node and edge features, $\mathbf{x}_v \in \mathbb{R}^f, \forall v \in V$ and $\mathbf{e}_{u,v} \in \mathbb{R}^{f'}, \forall (u, v) \in E$ representing atom types, formal charges,

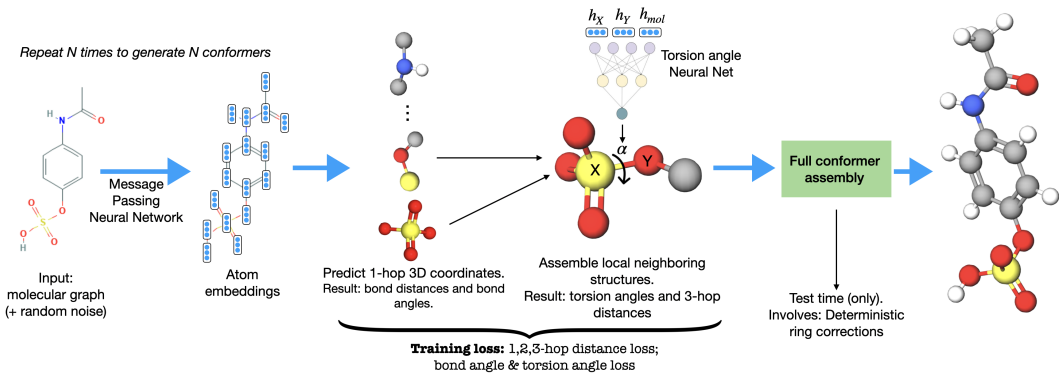

Figure 2: Overview of the GEOMOL model, which is SE(3)-invariant by design. Given a molecular graph, we first compute MPNNs atom embeddings. Next, we predict the local 3D structures (LS) of each non-terminal atom in a permutation invariant way, explicitly solving chirality. Third, for each bond connecting non-terminal vertices, we assemble the two LS by predicting a single torsion angle, avoiding overparameterization. Finally, the full conformer is assembled (only) at test time.

bond types, etc. For each molecular graph G, we have a variable-size set of low-energy ground truth 3D conformers $\{\mathcal{C}_l^*\}_l$ that we predict with a model $\{\mathcal{C}_k\}_k \stackrel{\text{def}}{=} \zeta(G)$. A conformer is a map $\mathcal{C} : V \to \mathbb{R}^3$ from graph nodes to 3D coordinates, but a simplified notation is $\mathbf{c}_v \in \mathbb{R}^3$ for $v \in V$. We use additional notations: $d(X, Y) = \|\mathbf{c}_X - \mathbf{c}_Y\|$ is the 3D distance between X and Y; $\angle XYZ$ is the counter-clockwise (CCW) angle $\angle \mathbf{c}_X \mathbf{c}_Y \mathbf{c}_Z$; $\angle(XYZ, XYT)$ is the CCW dihedral angle of the 2D planes $\mathbf{c}_X\mathbf{c}_Y\mathbf{c}_Z$ and $\mathbf{c}_X\mathbf{c}_Y\mathbf{c}_T$ (formula is in appendix C). We use the corresponding $\mathbf{c}_v^*, d^*(X, Y), \angle^* XYZ, \angle^*(XYZ, XYT)$ when manipulating a ground truth conformer.

Any conformer is defined up to a SE(3) transformation, i.e. any translation or rotation applied to the set $\{\mathbf{c}_v\}_{v \in V}$. A classic conformer distance function that satisfies this constraint is root-mean-square deviation of atomic positions (RMSD), computed by the Kabsch alignment algorithm [Kabsch, 1976].

2.1 GEOMOL high-level overview

Our approach, shown in fig. 2, comprises three steps. First, we predict the local 3D structure of each non-terminal atom, which we deem *local structure* (LS), by combining self-attention layers and MPNNs with deterministic corrections for chiral centers. Bond distances and bond angles are computed from the predicted LS. Next, we assemble all neighboring pairs of LSs by predicting the torsion angles and aligning them. Importantly, since LSs are fixed, it suffices to only predict a single value for the dihedral angle of each bond. Towards this goal, we develop a canonical representation of torsion angles via a local coordinate system defined SE(3)-equivariantly w.r.t. the full structure, which allows us to predict exactly the number of degrees of freedom. Finally, at test time, we assemble all predicted pairs of neighboring LSs to construct the full conformer, applying deterministic ring corrections. In order to generate diverse conformers, we append random Gaussian noise vectors to each initial node feature vector and use an optimal transport-based loss function for training.

2.2 Message passing neural networks (MPNNs)

Given an input graph G, an MPNN [Gilmer et al., 2017, Battaglia et al., 2018, Yang et al., 2019] computes node embeddings $\mathbf{h}_v \in \mathbb{R}^d, \forall v \in V$ using T layers of iterative message passing:

$$\mathbf{h}_u^{(t+1)} = \psi\left(\mathbf{h}_u^{(t)}, \sum_{v \in \mathcal{N}_u} \phi(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{e}_{u,v})\right), \quad \text{where } \mathbf{h}_v^{(0)} \stackrel{\text{def}}{=} concat[\mathbf{x}_v, \mathbf{z}_v],\ \mathbf{z}_v \sim \mathcal{N}(\mathbf{0}, s\mathbf{I}_d) \tag{1}$$

for each $t \in [0 \,..\, T-1]$, where $\mathcal{N}_u = \{v \in V | (u, v) \in E\}$, while ψ and ϕ are generic functions, e.g. implemented using multilayer perceptrons (MLP) or attention [Veličković et al., 2017]. Final node embeddings are obtained by the embedding of the last layer: $\mathbf{h}_v \stackrel{\text{def}}{=} \mathbf{h}_v^{(T)}, \forall v \in V$. Finally, we also compute a molecular embedding: $\mathbf{h}_{mol} \stackrel{\text{def}}{=} MLP(\sum_{v \in V} \mathbf{h}_v)$. We leave comparison with other MPNN variants for future work, e.g. Kipf and Welling [2017], Veličković et al. [2017], Hamilton et al. [2017], Xu et al. [2019].

2.3 Local structure (1-hop) prediction model

Following notations in fig. 3, for each non-terminal graph vertex $X \in V$ having n graph neighbors $\mathcal{N}_X = \{T_i\}_{i \in [1..n]}$, we predict its *local 3D structure* (LS), i.e. the relative 3D positions of all T_i, when X is centered in the origin. The generic model is a function $f(\mathbf{h}_{T_1}, \ldots, \mathbf{h}_{T_n}; \mathbf{h}_X) = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3 \times n}$ that, additionally, should satisfy permutation equivariance w.r.t. T_i's, namely, the 3D position of each neighbor T_i should not change regardless of the ordering of the X's neighbors:

$$f(\mathbf{h}_{T_{\pi(1)}}, \ldots, \mathbf{h}_{T_{\pi(n)}}; \mathbf{h}_X) = (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(n)}), \forall \pi \in S_n \tag{2}$$

Our choice is the encoder part of a transformer [Vaswani et al., 2017], without any positional encoding, thus satisfying permutation equivariance. This model takes as input the set $\{concat[\mathbf{h}_{T_i}, \mathbf{h}_X]; i \in [1..n]\}$ *in any order* and synchronously updates the n embeddings based on several transformer layers. The final layer projects the embeddings to 3 dimensions, resulting in a list $(\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3 \times n}$ having the exact same node order as the input list.

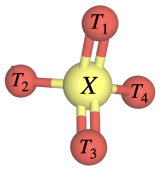

Figure 3: For each non-terminal atom X, we predict the relative 3D position of each of its graph neighbors, $\{T_i\}_{i \in [1..n]}$, in a permutation equivariant way.

Enforcing local consistency. We desire the LS model $f()$ to be *distance-consistent*, i.e. any bond distance $d(X, Y)$ is the same, no matter if it is computed from the LS of node X or of node Y. To achieve this, we use the above transformer just to compute bond directions (which will be aligned using a separate approach described in section 2.4), while we obtain the bond distances with a separate symmetric model. Concretely, let the above transformer $f()$ predict $(\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{3 \times n}$, while the final local 3D coordinates are $\mathbf{p}'_i \stackrel{\text{def}}{=} \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|} d_{GNN}(\mathbf{h}_X, \mathbf{h}_{T_i}), \forall i$, where each bond distance is predicted with a symmetric model $d_{GNN}(\mathbf{h}_X, \mathbf{h}_Y) \stackrel{\text{def}}{=} \text{softplus}(\psi(\mathbf{h}_X, \mathbf{h}_Y) + \psi(\mathbf{h}_Y, \mathbf{h}_X)), \forall (X, Y) \in E$, with the same shared ψ (e.g. an MLP). For notation simplicity, we will just use $\mathbf{p}_i$ instead of $\mathbf{p}'_i$.

What about SE(3) invariance? The above model is not SE(3)-invariant *per se*, but it is used as such. Namely, on one hand we compute SE(3)-invariant quantities: 1-hop distances $d(T_i, X)$, 2-hop distances $d(T_i, T_j)$, and bending angles $\angle T_i X T_j$. These will be compared to their ground-truth counterparts in the final loss, see section 2.5. On the other hand, the LS of adjacent graph nodes are assembled together for computing torsion angles or for building the full conformer at test time. This process is explicitly defined to be SE(3)-invariant as described in section 2.4.

Tetrahedral chiral corrections. When embedding the local neighborhood of a node in 3D space, one has to carefully account for tetrahedral chiral centers (fig. 4). Tetrahedral chirality is a common form of stereochemistry which restricts the 3D location of neighboring substituents of a central atom with four distinct neighbors; molecules which differ by a single tetrahedral chiral center, i.e. enantiomers, are mirror images of each other. Chirality heavily impacts some properties of small molecules–e.g. bioactivity. Existing MPNNs using only the molecular graph cannot distinguish chiral centers (fig. 4), but solutions exist [Pattanaik et al., 2020a]. Mathematically, enantiomers can be differentiated based on the oriented volume around the chiral center. That is, given the ordered set of neighbor 3D coordinates around the center, namely $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4 \in \mathbb{R}^3$, the sign of the volume of the tetrahedron formed by the neighbors is

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

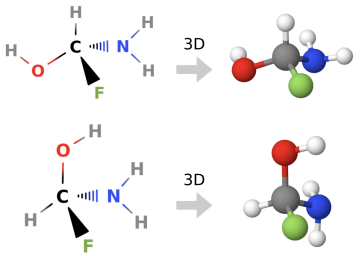

Figure 4: Chirality: even if the two shown graphs are isomorphic, they have distinct 3D structures that can be distinguished by the order of the carbon center's neighbors.

Enantiomeric structures always have opposite signs for the oriented volume [Crippen et al., 1988]. Since we generate local 3D structures directly, we can also use local 3D chiral descriptors to ensure the correct generation

of tetrahedral chiral centers. RDKit internally keeps track of these local chiral labels, denoted by CW/CCW labels (detailed in e.g. Pattanaik et al. [2020a]). Importantly, each local chiral label corresponds to a certain oriented volume (CW = +1 and CCW = -1). Thus, when generating an LS for a tetrahedral center, we calculate the oriented volume and check against the internal RDKit label. If it results in the incorrect oriented volume (i.e. the incorrect chiral center was generated), 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.

2.4 Torsion angle representation and local structure (LS) assembly

Once the LS of each atom/vertex is predicted, we assemble them in pairs corresponding to each non-terminal bond in the molecular graph. We describe this process for a bond connecting atoms X and Y, each having additional graph neighbors $\{T_i\}_{i\in[1..n]}$ and, resp., $\{Z_j\}_{j\in[1..m]}$. See fig. 5.

Torsion angle over-parameterization. We first note that, for any assembled bond XY (fig. 5, right), and $\forall i,k \in [1..n], \forall j,l \in [1..m]$, the dihedral angles $\angle(XYT_i, XYT_k)$ and $\angle(XYZ_l, XYZ_j)$ are fully determined by the LS of nodes X and Y, respectively, so they do not depend on torsion angles. Next, observe that there is exactly one torsion angle for any bond XY, given unique indexing of the neighbors. This happens because of the constraint on $\angle(XYT_i, XYZ_j)$ and $\angle(XYT_k, XYZ_l)$:

$$\angle(XYT_i, XYZ_j) = [\angle(XYT_k, XYZ_l) + \angle(XYT_i, XYT_k) + \angle(XYZ_l, XYZ_j)] (\mathrm{mod}\ 2\pi) \quad (3)$$

Thus, in order to avoid unnecessary over-parameterization, we should predict a single torsion angle α.

Torsion angle formulation. However, it is still unclear at this point how to define this unique angle in a canonical way that is both: i) permutation invariant w.r.t. the nodes in the set $\{T_i\}_{i\in[1..n]}$ and, respectively, in the set $\{Z_j\}_{j\in[1..m]}$, and ii) SE(3)-invariant w.r.t. the full 3D conformer.

Let $\Delta_{ij} \stackrel{\text{def}}{=} \angle(XYT_i, XYZ_j)$ and $\mathbf{s}_{ij} \stackrel{\text{def}}{=} \begin{bmatrix}\cos(\Delta_{ij})\\ \sin(\Delta_{ij})\end{bmatrix}$. Let $c_{ij} \in \mathbb{R}$ be real coefficients such that $\mathbf{s} \stackrel{\text{def}}{=} \sum_{i,j} c_{ij}\mathbf{s}_{ij} \in \mathbb{R}^2$ is not the null vector. Then, we define the torsion angle as[5]: $\alpha \stackrel{\text{def}}{=} atan2(\frac{\mathbf{s}}{\|\mathbf{s}\|})$. It is easy to see that this formulation satisfies both invariances claimed above. We further state (and prove in appendix A) that our proposed formulation gives a torsion angle uniquely determined by all local angles $\angle(YXT_i, YXT_k)$, $\angle(YXZ_j, YXZ_l)$ and by the true underlying torsion angle:

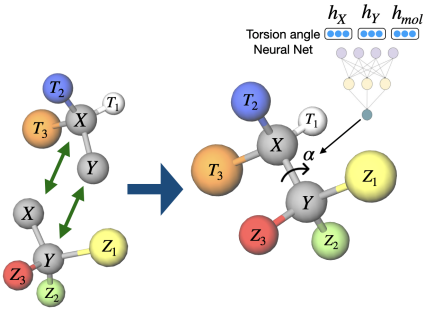

Figure 5: Assembly of the local structures of bonded atoms X and Y based on the predicted torsion angle.

Proposition 1. *Given 3D coordinates of nodes X, Y, T_i, Z_j and fixed weights $c_{ij} \in \mathbb{R}$ such that $\sum_{i,j} c_{ij}\mathbf{s}_{ij} \in \mathbb{R}^2$ is not the null vector, then $\alpha \stackrel{\text{def}}{=} atan2(\frac{\mathbf{s}}{\|\mathbf{s}\|})$ is unique, i.e. if we change the torsion angle of edge XY, then α will change. Formally, if we rotate the set of bonds $\{XT_i\}_i$ jointly around the line XY with the same angle γ, then α will be shifted with γ.*

How to set c_{ij}? Breaking symmetries. A simple solution is to choose $c_{ij} = 1, \forall i,j$. However, in some important cases, local symmetries may result in $\mathbf{s} = 0$. For example, this happens if, for some j, we have $\Delta_{ij} = \frac{2i\pi}{n} + ct., \forall i \in [1..n]$. One solution is to use different c_{ij} to differentiate between the different subgraphs rooted at different T_i (and similarly for Z_j). This is reminiscent of traditional group priorities used for distinguishing E/Z isomers. We devise a flexible solution: a differentiable real value function computed from the MPNN node embeddings as $c_{ij} = MLP(\mathbf{h}_{T_i} + \mathbf{h}_{Z_j}) \in \mathbb{R}$, with MLP being a network shared across all bonds and molecules. Note that we constrain $c_{ij} = c_{ji}$, thus guaranteeing that the same α is obtained if we swap X and Y (and their neighbors, respectively).

Final LS assembly for a single bond. We now describe the assembly process depicted in fig. 5. We first predict the LS of node X as in section 2.3, giving some $\mathbf{p}_X = \mathbf{0}, \mathbf{p}_Y, \mathbf{p}_{T_i} \in \mathbb{R}^3, \forall i \in [1..n]$,

[5] We define atan2 slightly different than standard: $atan2(r\cos(\alpha), r\sin(\alpha)) \stackrel{\text{def}}{=} \alpha, \forall \alpha \in [0, 2\pi), r \in \mathbb{R}^*_+$.

as well as the LS of node Y, giving $\mathbf{q}_Y = \mathbf{0}, \mathbf{q}_X, \mathbf{q}_{Z_j} \in \mathbb{R}^3, \forall j \in [1..m]$. By design, we have that $\|\mathbf{q}_X\| = \|\mathbf{p}_Y\|$. These two sets are currently not aligned. To achieve this, we first rotate the LS of X such that $\mathbf{p}_Y$ becomes $[\|\mathbf{p}_Y\| \quad 0 \quad 0]^\top$, while $\mathbf{p}_X$ remains $\mathbf{0}$. Next, we rotate and translate the LS of Y such that $\mathbf{q}_Y$ becomes $\mathbf{p}_Y$ and $\mathbf{q}_X$ becomes $\mathbf{p}_X = \mathbf{0}$. These two rotations have one degree of freedom each, which we set randomly. Exact formulas are in appendix B. Thus, the bond XY is now matched, but the torsional rotation is still arbitrary/random. The remaining step is to rotate the LS of X with an angle γ such that all dihedrals $\angle(XYT_i, XYZ_j)$ match their true counterparts. This is done by applying to all vectors $\mathbf{p}_{T_i}$ the same rotation of type: $\mathbf{H}_\gamma := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}$.

How to compute γ? The current dihedrals $\Delta_{ij}^{cur} \stackrel{\text{def}}{=} \angle^{cur}(XYT_i, XYZ_j)$ depend on the random torsional rotations from the previous assembly step. After applying the $\mathbf{H}_\gamma$ rotation, we obtain the new dihedral angles: $[\Delta_{ij}^{cur} - \gamma] \bmod 2\pi$ that should match the ground truth dihedral angles $\Delta_{ij}^* \stackrel{\text{def}}{=} \angle^*(XYT_i, XYZ_j)$. This is equivalently written as $\mathbf{s}_{ij}^* = \mathbf{A}_{ij}^{cur}\mathbf{s}_\gamma$, where $\mathbf{s}_\gamma \stackrel{\text{def}}{=} \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix}$ and $\mathbf{A}_{ij}^{cur} \stackrel{\text{def}}{=} \begin{bmatrix} \cos(\Delta_{ij}^{cur}) & \sin(\Delta_{ij}^{cur}) \\ \sin(\Delta_{ij}^{cur}) & -\cos(\Delta_{ij}^{cur}) \end{bmatrix}$. Let $\mathbf{s}^* \stackrel{\text{def}}{=} \sum_{i,j} c_{ij}\mathbf{s}_{ij}^*$ and $\mathbf{A}^{cur} \stackrel{\text{def}}{=} \sum_{i,j} c_{ij}\mathbf{A}_{ij}^{cur}$. The necessary condition for γ becomes $\mathbf{s}_\gamma = (\mathbf{A}^{cur})^\top \mathbf{s}^*$, which is also sufficient due to proposition 1. This implies it is enough to predict *only* the normalized $\begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix} \stackrel{\text{def}}{=} \frac{\mathbf{s}^*}{\|\mathbf{s}^*\|}$, from which γ follows. We predict α using a function commutative in X and Y (i.e. swapping X and Y does not change α):

$$\alpha = [\phi(\mathbf{h}_X, \mathbf{h}_Y, \mathbf{h}_{mol}) + \phi(\mathbf{h}_Y, \mathbf{h}_X, \mathbf{h}_{mol})] \bmod 2\pi \tag{4}$$

where ϕ is a neural network (e.g. MLP). Finally, $\mathbf{s}_\gamma = \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix} = \frac{1}{\|(\mathbf{A}^{cur})^\top \mathbf{s}^*\|} (\mathbf{A}^{cur})^\top \mathbf{s}^*$.

2.5 An optimal transport (OT) loss function for diverse conformer generation

Loss per single conformer. Assume first that we predict a single conformer $\mathcal{C}$. Based on all LS and torsion angle predictions, we deterministically compute all 1/2/3-hop distances and bond/torsion angles. If the corresponding ground truth conformer $\mathcal{C}^*$ is known, we feed those quantities into a negative log-likelihood loss, denote by $\mathcal{L}(\mathcal{C}, \mathcal{C}^*)$ and detailed in appendix D. Similar to Senior et al. [2020], we fit distances using normal distributions and angles using von Mises distributions. This is a much faster approach compared to habitual RMSD losses that compare full conformers.

Dealing with node symmetries. Our current formulation has difficulties distinguishing pairs of symmetric graph nodes that are less than 3 hops away, e.g. hydrogen groups. We address this using a tailored matching loss detailed in appendix D and exemplified in fig. 6.

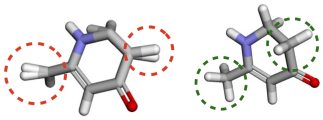

Figure 6: Before (left) and after (right) introducing a matching loss to distinguish symmetric graph nodes. Hydrogen predictions in both groups are visibly improved.

Total OT loss per ensemble of conformers. In practice, our model generates a set of conformers $\{\mathcal{C}_k\}_{k\in[1..K]}$ that needs to match a variable sized set of low-energy ground truth conformers, $\{\mathcal{C}_l^*\}_{l\in[1..L]}$. However, we do not know a priori the number L of true conformers or the matching between generated and true conformers. We also wish to avoid expensive and problematic adversarial training. Our solution is an OT-based, minimization-only, loss function:

$$\mathcal{L}^{ensemble} \stackrel{\text{def}}{=} \mathcal{W}_{\mathcal{L}(\cdot,\cdot)}(\{\mathcal{C}_k\}_k, \{\mathcal{C}_l^*\}_l) = \min_{\mathbf{T}\in\mathcal{Q}_{K,L}} \sum_{k,l} T_{kl}\mathcal{L}(\mathcal{C}_k, \mathcal{C}_l^*)$$

where $\mathbf{T}$ is the ***transport plan*** satisfying $\mathcal{Q}_{K,L} \stackrel{\text{def}}{=} \{\mathbf{T} \in \mathbb{R}_+^{K\times L} : \mathbf{T}\mathbf{1}_L = \frac{1}{K}\mathbf{1}_K, \mathbf{T}^T\mathbf{1}_K = \frac{1}{L}\mathbf{1}_L\}$. The minimization w.r.t. $\mathbf{T}$ is computed quickly using Earth Mover Distance and the POT library [Flamary and Courty, 2017].

Table 1: Results on the **GEOM-DRUGS** dataset. All models are without FF fine-tuning. "R" and "P" denote Recall and Precision. Note: OMEGA is an established commercial (C) software.

Models	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
	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. See caption of table 1.

Models	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
	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

2.6 Full conformer assembly at test time

Knowing all true LSs and torsion angles is, in theory, enough for a deterministic unique SE(3)-invariant reconstruction of the full conformer. However, in practice, these predictions might have small errors that accumulate, e.g. in rings. To mitigate this issue, we deterministically build the full conformer (only at test time) by first predicting a smoothed structure of (fused) rings separately, and then assembling the full conformer following any graph traversal order (any order gives the same conformer, so this procedure does not break the non-autoregressive behavior). We detail this step in appendix E.

3 Experiments

We empirically evaluate GEOMOL on the task of low-energy conformer ensemble generation for small and drug-like molecules. We largely follow the evaluation protocols of recent methods [Simm and Hernandez-Lobato, 2020, Xu et al., 2021], but also introduce new useful metrics.

Datasets & splits. We use two popular datasets: GEOM-QM9 [Ramakrishnan et al., 2014] and GEOM-DRUGS [Axelrod and Gomez-Bombarelli, 2020a]. Statistics and other details are in fig. 10 and in Mansimov et al. [2019]. Datasets are preprocessed as described in appendix G. We split them randomly based on molecules into train/validation/test (80%/10%/10%). At the end, for each dataset, we sample 1000 random test molecules as the final test set. Thus, the splits contain 106586/13323/1000 and 243473/30433/1000 molecules for GEOM-QM9 and GEOM-DRUGS, resp.

Baselines. We compare to established or recent baselines (discussed in section 1). **ETKDG/RDKit** [Riniker and Landrum, 2015] is likely the most popular open-source software, a stochastic DG-based method developed in the RDKit package. **OMEGA** [Hawkins et al., 2010, Hawkins and Nicholls, 2012, Friedrich et al., 2017], a rule-based method, is one of the most established commercial software, with more than a decade of continuous development. OMEGA and ETKDG are some of the fastest and best scaling existing approaches. Finally, we compare with the recent ML models of highest reported quality: **GraphDG** [Simm and Hernandez-Lobato, 2020] and **CGCF** [Xu et al., 2021].

Evaluation metrics. We follow prior work [Simm and Hernandez-Lobato, 2020, Xu et al., 2021] and use root-mean-square deviation of atomic positions (RMSD) to compare any two conformers. This is defined as the normalized Frobenius norm of the two corresponding matrices of 3D coordinates after being SE(3)-aligned a priori (using the Kabsch alignment algorithm [Kabsch, 1976]). Next, we introduce four types of metrics to compare two conformer ensembles, generated by a method,

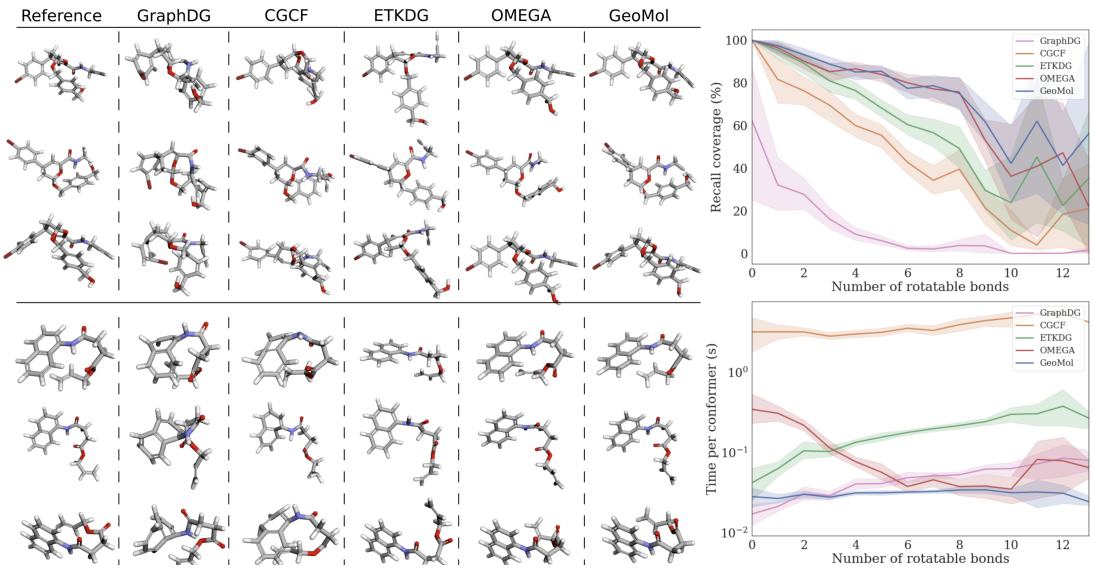

Figure 7: Left: Examples of generated structures. For every model, we show the best generated conformer, i.e. with the smallest RMSD to the shown ground truth. More examples are in appendix N. Right/top: Number of rotatable bonds per DRUGS test molecule versus COV Recall (95% confidence intervals). Right/bottom: conformer generation times for each model.

$\{\mathcal{C}_k\}_{k\in[1..K]}$, and ground truth, $\{\mathcal{C}_l^*\}_{l\in[1..L]}$. These metrics follow the established classification metrics of Precision and Recall and are defined for a given threshold $\delta > 0$ as:

$$\begin{aligned} \text{COV - R (Recall)} &\stackrel{\text{def}}{=} \frac{1}{L} \left|\{l \in [1..L] : \exists k \in [1..K], RMSD(\mathcal{C}_k, \mathcal{C}_l^*) < \delta\}\right| \\ \text{AMR - R (Recall)} &\stackrel{\text{def}}{=} \frac{1}{L} \sum_{l\in[1..L]} \min_{k\in[1..K]} RMSD(\mathcal{C}_k, \mathcal{C}_l^*) \end{aligned} \tag{5}$$

where AMR is "Average Minimum RMSD", COV is "Coverage", and *COV - P (Precision)* and *AMR - P (Precision)* are defined as in eq. (5), but with the generated and ground truth conformer sets swapped. The recall metrics measure how many of the ground truth conformers are correctly predicted, while the precision metrics indicate how many generated structures are of high quality. Specifically, in terms of recall, COV measures the percentage of correct generated conformers from the ground truth set (where a correct conformer is defined as one within an RMSD threshold of the true conformer), while AMR measures the average RMSD of each generated conformer with its closest groun truth match. Depending on the application, either of the metrics might be of greater interest. We follow Xu et al. [2021] and set $\delta = 0.5$Å for GEOM-QM9 and $\delta = 1.25$Å for GEOM-DRUGS.

Training and test details. For each input molecule having K ground truth conformers, we generate exactly $2K$ conformers using any of the considered methods. For GEOMOL, this is done by sampling different random noise vectors that are appended to node and edge features before the MPNN (eq. (1)). At train time, our model uses a standard deviation (std) s (see eq. (1)) of 5 for both GEOM-QM9 and GEOM-DRUGS. At test time, GEOMOL can use the same or different s values, depending on the downstream application, i.e. higher s results in more diverse conformers, while lower s gives more quality (better precision). For OMEGA, it is not possible to specify a desired number of conformers. So, we tune the RMSD threshold (which decides how many conformers to keep) such that the total generated conformers by OMEGA are approximately $2K$. For GEOM-QM9, this corresponds to no RMSD cutoff (i.e. OMEGA generates all possible conformers), and for GEOM-DRUGS, this corresponds to a cutoff of 0.7Å (meaning no two generated conformers will have a distance smaller than this cutoff). We discuss hyper-parameters and additional training details in appendix H.

Results & discussion. Results are shown in table 1 and table 2, and confidence intervals are in appendix I. As noted above, GEOMOL can be run with different noise std at test time, depending on

which metric the user is interested in. Even though OMEGA is an established commercial software with more than a decade of continuous development, our model remarkably frequently outperforms it. Note that OMEGA fails to generate any conformers for 7% of the QM9 test set (many of which include fused rings). Moreover, we also outperform the popular RDKit/ETKDG open-source model (except for AMR-P on QM9) and very recent ML models such as **GraphDG** and **CGCF**, sometimes by a large margin. For a qualitative insight, we show generated examples in fig. 7 and appendix N.

Additionally, we show in fig. 7 how COV Recall results are affected by the increasing number of rotatable bonds in the test molecule. As expected, having more rotatable bonds makes the problem harder, and this affects all baselines, but GEOMOL maintains a reasonable coverage even for more difficult molecules. Moreover, in appendix K and table 8 we show energy calculations of the generated conformers to support their plausibility. Additionally, results with energy-based relaxations are given in appendix J.

Running time. Fig. 7 shows conformer generation test running times. Our model is the fastest method from the considered baselines, being much faster than **CGCF** or **ETKDG/RDKit**. Moreover, GEOMOL scales favorably for molecules with increasing number of rotatable bonds.

4 Conclusion

We proposed GEOMOL, an end-to-end generative approach for molecular 3D conformation ensembles that explicitly models various molecular geometric aspects such as torsion angles or chirality. We expect that such differentiable structure generators will significantly impact small molecule conformer generation along with many related applications (e.g. protein-ligand binding), thus speeding up areas such as drug discovery. GEOMOL's full source code will be made publicly available.

Limitations & future work. A few current limitations are highlighted and left for future extensions (see also discussion in appendix O). First, our model does not currently support disconnected molecular graphs, e.g. ionic salts, but it can be applied to each connected component, followed by a 3D alignment. Next, our approach would benefit from explicit modeling of long distance interactions, especially for macrocycles or large molecules. This remains to be addressed in an efficient manner. Third, explicitly using ground truth energy values could further improve GEOMOL. Last, we look forward to fine-tune GEOMOL on applications such as generating molecular docking poses or descriptors for 4D QSAR.

Acknowledgments and Disclosure of Funding

OEG thanks Bracha Laufer, Tian Xie, Xiang Fu, Peter Mikhael, and the rest of the RB and TJ group members for their helpful comments and suggestions. LP thanks Camille Bilodeau and the rest of the WHG, KFJ, and CWC research groups for their useful discussions. We also thank Pat Walters, Simon Axelrod, and Rafael Gomez-Bombarelli for their insightful feedback as well as Minkai Xu and Shitong Luo for helping with running the CGCF model. Both OEG and LP are funded by the Machine Learning for Pharmaceutical Discovery and Synthesis consortium.

References

Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8(4): 292–301, 2019.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020a.

Simon Axelrod and Rafael Gomez-Bombarelli. Molecular machine learning with conformer ensembles. *arXiv preprint arXiv:2012.08452*, 2020b.

Andrew J Ballard, Stefano Martiniani, Jacob D Stevenson, Sandeep Somani, and David J Wales. Exploiting the potential energy landscape to sample free energy. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(3):273–289, 2015.

Arghya Barman, Bruce Batiste, and Donald Hamelberg. Pushing the limits of a molecular mechanics force field to probe weak ch$\cdots\pi$ interactions in proteins. *Journal of chemical theory and computation*, 11(4):1854–1863, 2015.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Lucian Chan, Garrett M Morris, and Geoffrey R Hutchison. Understanding conformational entropy in small molecules. *Journal of Chemical Theory and Computation*, 17(4):2099–2106, 2021.

Gordon M Crippen, Timothy F Havel, et al. *Distance geometry and molecular conformation*, volume 74. Research Studies Press Taunton, 1988.

Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of medicinal chemistry*, 59(9):4035–4061, 2016.

Rémi Flamary and Nicolas Courty. Pot python optimal transport library. *GitHub: https://github.com/rflamary/POT*, 2017.

Nils-Ole Friedrich, Christina de Bruyn Kops, Florian Flachsenberg, Kai Sommer, Matthias Rarey, and Johannes Kirchmair. Benchmarking commercial conformer ensemble generators. *Journal of chemical information and modeling*, 57(11):2719–2728, 2017.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pages 3419–3430. PMLR, 2020.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

Stefan Grimme. Exploration of chemical compound, conformer, and reaction space with metadynamics simulations based on tight-binding quantum chemical calculations. *Journal of chemical theory and computation*, 15(5):2847–2862, 2019.

Cristiano RW Guimaraes, Alan M Mathiowetz, Marina Shalaeva, Gilles Goetz, and Spiros Liras. Use of 3d properties to characterize beyond rule-of-5 property space for passive permeation. *Journal of chemical information and modeling*, 52(4):882–890, 2012.

Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

Thomas A Halgren. Mmff vi. mmff94s option for energy minimization studies. *Journal of Computational Chemistry*, 20(7):720–729, 1999.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

Timothy F Havel. Distance geometry: Theory, algorithms, and chemical applications. *Encyclopedia of Computational Chemistry*, 120:723–742, 1998.

Timothy F Havel, Gordon M Crippen, Irwin D Kuntz, and Jeffrey M Blaney. The combinatorial distance geometry method for the calculation of molecular conformation ii. sample problems and computational statistics. *Journal of theoretical biology*, 104(3):383–400, 1983a.

Timothy F Havel, Irwin D Kuntz, and Gordon M Crippen. The combinatorial distance geometry method for the calculation of molecular conformation. i. a new approach to an old problem. *Journal of theoretical biology*, 104(3):359–381, 1983b.

Paul CD Hawkins. Conformation generation: the state of the art. *Journal of Chemical Information and Modeling*, 57(8):1747–1756, 2017.

Paul CD Hawkins and Anthony Nicholls. Conformer generation with omega: learning from the data set and the analysis of failures. *Journal of chemical information and modeling*, 52(11):2919–2936, 2012.

Paul CD Hawkins, A Geoffrey Skillman, Gregory L Warren, Benjamin A Ellingson, and Matthew T Stahl. Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of chemical information and modeling*, 50(4):572–584, 2010.

John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations*, 2018.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Ilana Y Kanal, John A Keith, and Geoffrey R Hutchison. A sobering assessment of small-molecule force field methods for low energy conformer predictions. *International Journal of Quantum Chemistry*, 118(5):e25512, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

Ashutosh Kumar and Kam YJ Zhang. Advances in the development of shape similarity methods and their application in drug discovery. *Frontiers in chemistry*, 6:315, 2018.

David Lagorce, Tania Pencheva, Bruno O Villoutreix, and Maria A Miteva. Dg-ammos: A new tool to generate 3d conformation of small molecules using d istance g eometry and a utomated m olecular m echanics o ptimization for in silico s creening. *BMC Chemical Biology*, 9(1):1–10, 2009.

Dominik Lemm, Guido Falk von Rudorff, and O Anatole von Lilienfeld. Energy-free machine learning predictions of *ab initio* structures. *arXiv preprint arXiv:2102.02806*, 2021.

Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1):1–13, 2019.

Mark McGann. Fred pose prediction and virtual screening accuracy. *Journal of chemical information and modeling*, 51(3):578–596, 2011.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), 2019.

Lagnajit Pattanaik, Octavian E Ganea, Ian Coley, Klavs F Jensen, William H Green, and Connor W Coley. Message passing networks for molecules with tetrahedral chirality. *arXiv preprint arXiv:2012.00094*, 2020a.

Lagnajit Pattanaik, John B Ingraham, Colin A Grambow, and William H Green. Generating transition states of isomerization reactions with deep learning. *Physical Chemistry Chemical Physics*, 22 (41):23618–23626, 2020b.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574, 2015.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Kristof T Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

Christof H Schwab. Conformations and 3d pharmacophore searching. *Drug Discovery Today: Technologies*, 7(4):e245–e253, 2010.

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

Jihyun Shim and Alexander D MacKerell Jr. Computational ligand-based rational design: role of conformational sampling and force fields in model development. *MedChemComm*, 2(5):356–370, 2011.

Gregor Simm and Jose Miguel Hernandez-Lobato. A generative model for molecular distance geometry. In *International Conference on Machine Learning*, pages 8949–8958. PMLR, 2020.

Freyr Sverrisson, Jean Feydy, Bruno Correia, and Michael Bronstein. Fast end-to-end learning on protein surfaces. *bioRxiv*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jitender Verma, Vijay M Khedkar, and Evans C Coutinho. 3d-qsar in drug design-a review. *Current topics in medicinal chemistry*, 10(1):95–115, 2010.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations (ICLR)*, 2019.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. *International Conference on Learning Representations (ICLR)*, 2021.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370–3388, 2019.

A Proof of proposition 1

Proof. Let's assume we apply a random CCW torsion rotation of angle $\gamma \in (0, 2\pi)$ around bond XY to all bonds XT_i. We will prove that the resulting α will be shifted exactly by γ. Let us denote the current dihedrals after the γ rotation by $\Delta_{ij}^{cur} = \angle^{cur}(XYT_i, XYZ_j) \bmod 2\pi$. Also, denote by $\Delta_{ij} = \angle(XYT_i, XYZ_j) \bmod 2\pi$ the dihedral angles before rotation. We have the relation $\Delta_{ij} = [\Delta_{ij}^{cur} + \gamma] \bmod 2\pi, \forall i, j$, or, in matrix form:

$$\mathbf{s}_{ij} = \mathbf{A}_\gamma \mathbf{s}_{ij}^{cur} \tag{6}$$

where $\mathbf{A}_\gamma := \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix}$, $\mathbf{s}_{ij} := \begin{bmatrix} \cos(\Delta_{ij}) \\ \sin(\Delta_{ij}) \end{bmatrix}$ and $\mathbf{s}_{ij}^{cur} := \begin{bmatrix} \cos(\Delta_{ij}^{cur}) \\ \sin(\Delta_{ij}^{cur}) \end{bmatrix}$.

We know use $\mathbf{s} := \sum_{i,j} c_{ij}\mathbf{s}_{ij} \in \mathbb{R}^2$ and $\mathbf{s}^{cur} := \sum_{i,j} c_{ij}\mathbf{s}_{ij}^{cur} \in \mathbb{R}^2$. From the above relation, we obtain that $\frac{\mathbf{s}}{\|\mathbf{s}\|} = \mathbf{A}_\gamma \frac{\mathbf{s}^{cur}}{\|\mathbf{s}^{cur}\|}$. Writing $\frac{\mathbf{s}}{\|\mathbf{s}\|} = \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}$ and $\frac{\mathbf{s}^{cur}}{\|\mathbf{s}^{cur}\|} = \begin{bmatrix} \cos(\alpha^{cur}) \\ \sin(\alpha^{cur}) \end{bmatrix}$ we obtain that $\alpha = \gamma + \alpha^{cur}$ which concludes our proof. □

A note on collinear 3D points. In the main text, we always assumed that dihedral angles $\Delta_{ij} \stackrel{\text{def}}{=} \angle(XYT_i, XYZ_j), \forall i \in [1..n], \forall j \in [1..m]$ exist. However, any such angle will not be defined if X and Y are collinear with either T_i or Z_j. In such cases, we identify those neighbor nodes that are predicted to be collinear with X and Y (solely based on their respective LS), and then remove them from the computation of $\mathbf{A}^{cur}$ and $\mathbf{s}_\gamma$. In the extreme case when these collinear nodes are the only neighbors, the torsion angle is not defined by our formulation. We leave this isolated case to be solved in future extensions of our work.

B Initial assembly of the LSs of the endpoints of a bond XY

We detail here the formulae used in section section 2.4. Let the (predicted) LS of node X be $\mathbf{p}_X = \mathbf{0}, \mathbf{p}_Y, \mathbf{p}_{T_1}, \ldots, \mathbf{p}_{T_n} \in \mathbb{R}^3$, as in section 2.3. Similarly, let the LS of node Y be $\mathbf{q}_Y = \mathbf{0}, \mathbf{q}_X, \mathbf{q}_{Z_1}, \ldots, \mathbf{q}_{Z_m} \in \mathbb{R}^3$. By design of section 2.3, we have that $\|\mathbf{q}_X\| = \|\mathbf{p}_Y\|$. These sets of 3D points can be aligned by applying any SE(3) transformation to each of them. To achieve alignment, we first want to match the X and Y points.

For this, we first rotate the LS of X such that $\mathbf{p}_Y$ becomes $\begin{bmatrix} \|\mathbf{p}_Y\| \\ 0 \\ 0 \end{bmatrix}$ and $\mathbf{p}_X$ remains $\mathbf{0}$. Next, we rotate and translate the LS of Y such that $\mathbf{q}_Y$ becomes $\mathbf{p}_Y$ and $\mathbf{q}_X$ becomes $\mathbf{0}$. More concretely, let $\eta_X \in \mathbb{R}^3$ be any random unit-norm vector orthogonal to $\mathbf{p}_Y$. We find a rotation matrix $\mathbf{H}_{X,Y} = \begin{bmatrix} - & \mathbf{h}_1^\top & - \\ - & \mathbf{h}_2^\top & - \\ - & \mathbf{h}_3^\top & - \end{bmatrix} \in \mathbb{R}^{3\times 3}$ such that $\mathbf{H}_{X,Y}\mathbf{p}_Y = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$, with $a > 0$, and $\mathbf{H}_{X,Y}\eta_X = \begin{bmatrix} b \\ c \\ 0 \end{bmatrix}$, where $b, c \in \mathbb{R}$. This is solved as:

$$\mathbf{h}_1 = \frac{\mathbf{p}_Y}{\|\mathbf{p}_Y\|}; \quad \mathbf{h}_3 = \frac{\mathbf{p}_Y \times \eta_X}{\|\mathbf{p}_Y \times \eta_X\|}; \quad \mathbf{h}_2 = -(\mathbf{h}_1 \times \mathbf{h}_3)$$

We now rotate the LS of X by doing: $\mathbf{p}'_{T_i} \stackrel{\text{def}}{=} \mathbf{H}_{X,Y}\mathbf{p}_{T_i}, \forall i \in [1..n]$, $\mathbf{p}'_Y \stackrel{\text{def}}{=} \mathbf{H}_{X,Y}\mathbf{p}_Y$. We apply a similar procedure for the LS of Y, computing $\mathbf{q}'_{Z_j} \stackrel{\text{def}}{=} \mathbf{H}_{Y,X}\mathbf{q}_{Z_j}, \forall j \in [1..m]$ and $\mathbf{q}'_X := \mathbf{H}_{Y,X}\mathbf{q}_X$. We now reflect the LS of Y w.r.t. the first and second coordinate by doing $\mathbf{q}'_{Z_j} \leftarrow \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{q}'_{Z_j} + \mathbf{p}'_Y$, which centers X in the origin, i.e. $\mathbf{q}'_X = \mathbf{0}$, and aligns the two vectors of Y, i.e. $\mathbf{q}'_Y = \mathbf{p}'_Y$.

C Dihedral angle formula

The CCW dihedral angle between two intersecting half-planes ABC and ABD with common points A,B is computed as[6]

$$\angle(ABC, ABD) = atan2(\|\mathbf{b}_2\|\langle \mathbf{b}_1, \mathbf{b}_2 \times \mathbf{b}_3\rangle, \langle \mathbf{b}_1 \times \mathbf{b}_2, \mathbf{b}_2 \times \mathbf{b}_3\rangle) \tag{7}$$

where $\mathbf{b}_1 := \mathbf{s}_A - \mathbf{s}_C, \mathbf{b}_2 := \mathbf{s}_B - \mathbf{s}_A, \mathbf{b}_3 := \mathbf{s}_D - \mathbf{s}_B$.

D Details of the loss function

Assume, for a molecular graph $G = (V, E)$, that one predicted/generated conformer using one model is $\mathcal{C}$, while the corresponding ground truth conformer is $\mathcal{C}^*$. We use notations in section 2. E is the set of edges in graph G. For every chain of 3 edges $(u, v), (v, w), (w, y) \in E$, we define $\angle(uvwy) \stackrel{\text{def}}{=} \angle(uvw, vwy)$.

Similar to AlphaFold [Senior et al., 2020], we fit distances using normal distributions and angles using von Mises distributions. The resulting negative log-likelihood loss averages over same types of terms, being formally written as:

$$\begin{aligned}
\mathcal{L}(\mathcal{C}, \mathcal{C}^*) \stackrel{\text{def}}{=} & \xi_1 \cdot \frac{1}{\#\{(u,v) \in E\}} \sum_{\{(u,v)\in E\}} (d(u,v) - d^*(u,v))^2 \\
& + \xi_2 \cdot \frac{1}{\#\{u, v : \text{2-hops away}\}} \sum_{\{u,v:\text{2-hops away}\}} (d(u,v) - d^*(u,v))^2 \\
& + \xi_3 \cdot \frac{1}{\#\{u, v : \text{3-hops away}\}} \sum_{\{u,v:\text{3-hops away}\}} (d(u,v) - d^*(u,v))^2 \\
& - \xi_4 \cdot \frac{1}{\#(u,v) \in E, (v,w) \in E} \sum_{(u,v)\in E, (v,w)\in E} \cos(\angle uvw - \angle^* uvw) \\
& - \xi_5 \cdot \frac{1}{\#(u,v), (v,w), (w,y) \in E} \sum_{(u,v),(v,w),(w,y)\in E} \cos(\angle(uvwy) - \angle^*(uvwy))
\end{aligned} \tag{8}$$

where $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5 > 0$ are hyperparameters tuned on the validation set, with ξ_1, ξ_2, ξ_3 representing inverse standard deviations of the respective normal distributions, while ξ_4, ξ_5 being the measures of concentration of the corresponding von Mises distributions.

Dealing with node symmetries. A matching loss function. So far, we haven't tackled the following difficulty: symmetric graph nodes that are less than 3 hops away are indistinguishable by MPNNs in general, and by our current model in particular, even if we append initial random noise feature vectors. However, these nodes have distinct distances and angles to other nodes in 3D space. Examples are hydrogen groups as in fig. 8. The difficulty in our case comes from the fact that the MPNN has lost the true matching between the graph nodes and the respective 3D points of the ground truth conformer. Our model is able to differentiate symmetric nodes because of the initial random noise vectors appended to each atom and feature vector, but a consistent matching with the ground truth conformer nodes is lost, and we seek to recover it.

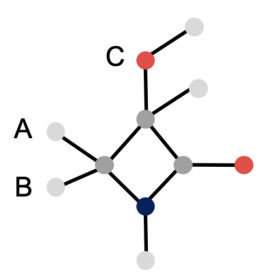

Figure 8: Nodes A and B are symmetric in this example graph, but not when placed in the 3D space, as node A will be spatially closer to node C than node B. Such cases require a special treatment.

We here address this problem only for symmetric terminal nodes attached to the same common neighbor. We leave

[6] See `https://en.wikipedia.org/wiki/Dihedral_angle`.

further extensions to more general symmetries for future work. Concretely, let $X_1, \ldots, X_T$ be identical atoms of degree 1, each being connected to the same common node Y in the molecular graph. Let $\mathbf{c}_1, \ldots, \mathbf{c}_T \in \mathbb{R}^3$ be their predicted 3D coordinates in one generated conformer $\mathcal{C} : V \to \mathbb{R}^3$. Let also $\mathbf{c}_1^*, \ldots, \mathbf{c}_T^* \in \mathbb{R}^3$ be their corresponding 3D coordinates of some ground truth conformer $\mathcal{C}^* : V \to \mathbb{R}^3$. These coordinates are then used for the computation of different loss terms in eq. (8).

We propose a new loss function based on eq. (8) that replaces it:

$$\mathcal{L}^{perm}(\mathcal{C}, \mathcal{C}^*) \stackrel{\text{def}}{=} \min_{\pi \in S_T} \mathcal{L}(\mathcal{C}, \mathcal{C}_\pi^*) \tag{9}$$

where $\mathcal{C}_\pi^* : V \to \mathbb{R}^3$ is defined as $\mathcal{C}_\pi^*(v) = \mathcal{C}^*(v), \forall v \in V \setminus \{X_1, \ldots, X_T\}$ and $\mathcal{C}_\pi^*(X_i) = \mathcal{C}^*(X_{\pi(i)}), \forall i \in [1..T]$. In a nutshell, this new loss function searches all possible permutations of the symmetric nodes $X_1, \ldots, X_T$ in order to find the "best matching" (based on the loss value) with the corresponding points in the ground truth conformer $\mathcal{C}^*$. The procedure above is applied for all groups of symmetric graph nodes that are of degree 1 and attached to a common node in the molecular graph. The loss in eq. (9) does not significantly affect the speed of our method (as we show in the experimental section) due to the fact that most groups of symmetric atoms contain at most 3 nodes. However, it positively affects the quality of the resulting conformers as exemplified in fig. 6.

OT Loss. During training, the matrix $\mathbf{T}$ in section 2.5 is computed for the forward pass using the Earth Mover Distance (EMD) function from the POT library [Flamary and Courty, 2017] and kept fixed during backpropagation. The EMD computation cannot be parallelized in mini-batches in the current version of the library, but everything else is batch-parallelizable in our model (e.g. computation of $\mathcal{L}(\mathcal{C}, \mathcal{C}^*)$).

E Details of the full conformer assembly procedure at test time

The training stage happens without assembling the full conformer. However, we desire to generate the full 3D conformer for test and inference time. This can be done by repeatedly applying the assembly operation described in section 2.4. We assemble the 3D coordinates of the atoms one by one, following a fixed arbitrary graph traversal, say Breadth First Search (BFS). Our procedure described here is deterministic, and any graph traversal will lead to the same final conformer (up to an SE(3) transformation).

The key step is assembling two sets of 3D points: the set containing the LS of node X (and possibly other atom coordinates added in previous steps), denoted as $S_X := \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subset \mathbb{R}^3$, and the set containing the LS of node Y, denoted as $S_Y := \{\mathbf{q}_1, \ldots, \mathbf{q}_m\} \subset \mathbb{R}^3$. Assume that X and Y are connected by a bond/edge. Also assume that X is connected to nodes T_i, while Y is connected to nodes Z_j as denoted in fig. 5. We make the clarification that $\mathbf{p}_{T_i} \in S_X, \mathbf{p}_Y \in S_X, \mathbf{p}_X \in S_X$ and $\mathbf{q}_{Z_j} \in S_Y, \mathbf{q}_Y \in S_Y, \mathbf{q}_X \in S_Y$ and we expect that $\|\mathbf{p}_X - \mathbf{p}_Y\| = \|\mathbf{q}_X - \mathbf{q}_Y\|$ by our design of this model.

To assemble the sets S_X and S_Y, we follow the steps described in section 2.4 and appendix B, but using only points in the LS of X from the set S_X together with only points of the LS of Y from S_Y to compute the torsion angle γ of the bond XY. That is, the torsion angle is computed only based on the graph neighbors of nodes X and Y. However, all points in S_X and S_Y are used for all the updates of the 3D coordinates. In the end, we obtain assembled set of points that satisfy $\mathbf{q}_Y = \mathbf{p}_Y = \begin{bmatrix} d_{GNN}(\mathbf{h}_X, \mathbf{h}_Y) \\ 0 \\ 0 \end{bmatrix}, \mathbf{q}_X = \mathbf{p}_X = \mathbf{0}$. Finally, we just merge the two sets S_X and S_Y into their union.

Dealing with graph cycles. The assembly operation might lead to inconsistencies when a node is reached the second time after a cycle traversal. We adopt a deterministic approach to tackle this issue. We first fix one BFS traversal of the graph and assemble the LS in the order given by this traversal. If the current node is not part of a cycle, then we can perform the assembling as described before. However, when we first encounter a node that is part of a cycle of nodes $X_1, X_2, \ldots, X_n$, we will jointly compute all the 3D coordinates of this cycle and attach the entire ring structure to the current partial conformer. Concretely, for every $i \in \{1, \ldots, n\}$, we assemble the LS of nodes

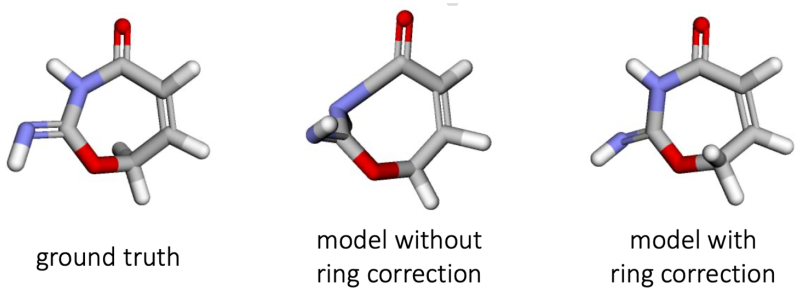

Figure 9: Effect of two assembly techniques. Left: ground truth structure. Middle: our assembly procedure is applied on a random spanning tree of the molecular graph. Right: our described ring correction/smoothing algorithm, leading to a visibly improved structure.

$i, i+1, \ldots n, 1, 2, \ldots i-2$ resulting in the 3D points $\mathbf{x}_{i-1}, \mathbf{x}_i\mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-2}, \mathbf{x}'_{i-1}$, where node $i-1$ has two 3D points $\mathbf{x}_{i-1}$ and $\mathbf{x}'_{i-1}$ computed based on the LS of nodes i and $i-2$ respectively. Next, we average the two 3D points of node $i-1$ into a single vector denoted as $\mathbf{x}_{i-1} \in \mathbb{R}^3$. We denote the resulting list of 3D points as $S_i = \{\mathbf{x}_{i-1}, \mathbf{x}_i\mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-2}\}$. So, we have obtained n lists of 3D points, $S_i, \forall i \in [1..n]$, that each represents the 3D conformer of the same cycle when the assembling is done using all nodes except one, in turn. We align all S_i together using Kabsch algorithm [Kabsch, 1976] [7], followed by averaging all vectors of the same cycle node, for each node in the cycle, thus obtaining a final smoothed set of 3D coordinates for this respective cycle. We call this procedure "ring correction" or "ring smoothing". We additionally note that, when aligning two sets S_i and S_j, we also have to align the non-cycle points previously assembled. Finally, after all 3D vectors of a cycle are obtained, we continue to assemble in the order given by the BFS traversal. We show an example of the effect of this procedure in fig. 9.

F Datasets statistics

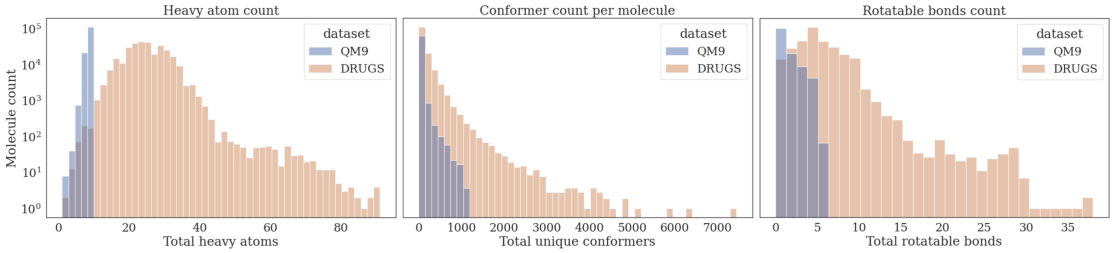

Figure 10: Datasets statistics.

We here show several dataset statistics in fig. 10, but point the interested reader to other resources describing these datasets, e.g. Mansimov et al. [2019], Axelrod and Gomez-Bombarelli [2020a], Ramakrishnan et al. [2014].

Datasets generation details. The GEOM dataset is generated with semi-empirical tight-binding DFT (GFN2-xTB) with the CREST software Grimme [2019]. However, such semi-empirical calculations can be relatively slow in practice–espeically for larger molecules–which is one of the main motivations of ML models for conformer ensemble prediction.

[7] `https://en.wikipedia.org/wiki/Kabsch_algorithm`

G Data preprocessing

We featurize each .pickle file in the datasets, which correspond to a single molecule and its respective conformers, to PyTorch Geometric data objects. First, we ensure the provided SMILES strings can be processed by RDKit, and we discard the few molecules that fail to meet this criteria. We additionally discard molecules with disconnected fragments (i.e. contains a "." in the SMILES) and without any dihedral angles (i.e. fails to match a "$[*] \sim [*] \sim [*] \sim [*]$" SMARTS pattern). Finally, some conformers in the original dataset may have reacted during the original data generation process. We filter these conformers by inferring the SMILES from the 3D structure, canonicalizing the SMILES with RDKit, and checking this SMILES with the SMILES reported by the dataset. Any conformers who fail this check are discarded as reacted conformers. We detail the atom and bond features used for the final GEOM-DRUGS model in Table 3. Note that the GEOM-QM9 model uses the same features but restricts the possible atom identities to H, C, N, O, and F.

Table 3: Atom and bond features

Atom features			
Indices	Description	Options	Type
0-34	atom identity	H, Li, B, C, N, O, F, Na, Mg, Al, Si, P, S, Cl, K, Ca, V, Cr, Mn, Cu, Zn, Ga, Ge, As, Se, Br, Ag, In, Sb, I, Gd, Pt, Au, Hg, Bi	one-hot
35	atomic number	$\mathbb{Z}_{>0}$	value
36	aromaticity	true, false	one-hot
37-44	degree	0, 1, 2, 3, 4, 5, 6, other	one-hot
45-50	hybridization	sp, sp^2, sp^3, sp^3d, sp^3d^2, other	one-hot
51-58	implicit valence	0, 1, 2, 3, 4, 5, 6, other	one-hot
59-62	formal charge	-1, 0, 1, other	one-hot
63-69	presence of atom in ring of size x	3, 4, 5, 6, 7, 8, other	k-hot
70-74	number of rings atom is in	0, 1, 2, 3, other	one-hot
Bond features			
Indices	Description	Options	Type
0-3	bond type	single, double, triple, aromatic	one-hot

H Additional training details and hyperparameters

We run each model for a fixed 250 epochs with early stopping and keep the model with the best validation performance. This strategy determines the final hyperparameters chosen for the model, shown in table 4. We additionally tune the standard deviation s on a sample of the validation set to maximize recall coverage or precision coverage (depending on the user's interest), and we show the corresponding test scores in our tables. To adjust the learning rate during training, we use a plateau scheduler which automatically decays the learning rate when improvement has stalled, using a decay factor of 0.7 and a patience of 5 epochs. All models use the same seed to initialize network weights, ensuring consistency between runs. During training time, we randomly sample a fixed number of ground truth conformers from each molecule. This fixed number is a hyperparameter in our optimization, which we set to 10 for GEOM-QM9 and 20 for GEOM-DRUGS. This value is not relevant during test time, as the model simply generates the number of conformers requested by the user.

I Confidence intervals

We report in table 5 confidence intervals for GEOMOL on both datasets during test time. The same trained model is run for 3 times for a given test molecule with K ground truth conformers, where one run means generating $2K$ conformers.

Table 4: Hyerparameter choices

Hyperparameter	Final choice (QM9/DRUGS)
True conformers sampled	10/20
Model conformers generated	10/20
Model dimension	25
Random vector dimension	10
Random vector standard deviation s	5
MPNN 1 depth	3
MPNN 1 number of layers	2
MPNN 2 depth	3
MPNN 2 number of layers	2
LS self-attention encoder heads	2
LS coordinate MLP layers	2
LS distance MLP layers	1
h_{mol} MLP layers	1
α MLP layers	2
c_{ij} MLP layers	1
Batch size	16
Learning rate	1e-3

Table 5: Confidence intervals for GEOMOL with the default noise $s = 5$.

	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
Models	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GEOMOL DRUGS	82.43 ± 0.26	95.10 ± 0.25	0.8626 ± 0.0020	0.8374 ± 0.0037	78.52 ± 0.02	94.40 ± 0.20	0.9336 ± 0.0015	0.8567 ± 0.0030
GEOMOL QM9	91.52 ± 0.36	100.00 ± 0.00	0.2254 ± 0.0012	0.1938 ± 0.0024	86.71 ± 0.29	100.00 ± 0.00	0.2702 ± 0.0007	0.2413 ± 0.0030

J Results with force field optimization

We also show results in table 6 and table 7 for all models after additional fine-tuning with FF energy minimization. We used MMFF94s Merck Molecular Force Field [Halgren, 1996, 1999]). Note that the best GEOMOL models here use $s = 1$ during training rather than $s = 5$.

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

	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
Models	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GraphDG *(ML)*	85.33	100.00	0.859	0.831	62.65	69.45	1.162	1.121
CGCF *(ML)*	**91.25**	**100.00**	0.723	0.702	63.86	70.43	1.096	1.055
RDKit/ETKDG	77.22	87.62	0.886	0.837	75.98	93.86	0.911	0.803
OMEGA *(C)*	80.82	94.74	0.840	0.755	81.79	**100.00**	0.804	0.684
GEOMOL ($s = 0.25$)	80.73	92.01	0.863	0.809	**86.12**	**100.00**	**0.751**	**0.616**
GEOMOL ($s = 3.00$)	**91.34**	**100.00**	**0.683**	**0.663**	79.64	92.46	0.841	0.756

K Energy calculations

To gauge the plausibility of generated conformers, we compute the energies as defined by the MMFF force field within RDKit for conformers generated with ML-based methods before force field fine tuning. We note that an improved demonstration would calculate energies at the GFN2-xTB semiempirical level of theory, since the training data were generated with this method, but we decided to use a force field instead, which should retain the qualitative trends. Table 8 illustrates these results, which support the results from table 1 and table 2. The energy values from GeoMol are the lowest among the ML methods, indicating greater stability of generated conformers, especially for the druglike molecules.

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

Models	COV - R (%) ↑		AMR - R (Å) ↓		COV - P (%) ↑		AMR - P (Å) ↓	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GraphDG *(ML)*	88.70	100.00	0.210	0.165	90.14	**100.00**	0.185	0.129
CGCF *(ML)*	75.45	100.00	0.313	0.246	50.29	50.00	0.518	0.520
RDKit/ETKDG	83.48	100.00	0.219	0.172	89.78	**100.00**	**0.160**	**0.116**
OMEGA *(C)*	85.73	100.00	**0.177**	**0.126**	83.08	**100.00**	0.224	0.186
GEOMOL ($s = 5.00$)	**89.37**	100.00	0.201	0.157	**91.92**	**100.00**	0.173	0.124

Table 8: MMFF energy calculations for ML conformer generators

Model	**Mean/Median (kcal/mol)**	
	QM9	DRUGS
GeoMol	17.45/17.56	83.72/77.52
GraphDG	26.32/24.44	64350.58/61732.49
CGCF	86.71/87.02	1380.82/556.78

L Compute details

We train GEOMOL using a single Nvidia Volta V100 GPU and eight Intel Xeon Gold 6248 CPUs. The exact number of hours depends on the cluster load and on early stopping, but training GEOMOL for the full 250 epochs takes approximately 1.8 days for the QM9 dataset and 6.1 days for the DRUGS dataset.

M Additional coverage results

We additionally show how the Recall Coverage varies with the threshold δ in fig. 11. Larger thresholds allow small errors, but penalize large errors, while smaller thresholds penalize both small and large errors.

N Additional examples

We show additional examples of generated conformers from the GEOM-QM9 test set in fig. 12 and from the GEOM-DRUGS test set in fig. 13.

O Additional discussion

O.1 Purpose of conformer generation

The purpose of conformer generation in the context of this work warrants some discussion. As stated in the main text, this purpose is heavily application-dependent. The current method is very helpful for computing the properties of drug-like molecules at low temperature, where it is necessary to generate a set of conformers that represents those with high Boltzmann populations. It can also be used for small molecules, but if the molecules are very small, it may be possible to identify all the important conformers by exhaustive search. At very high temperatures, where a large number of conformers are energetically accessible, it may be more convenient to represent the potential energy surface as a sum of hindered rotor potentials. Yet another common application for 3D conformer generation is the discovery of docked poses and 4D QSAR, where speed of the conformer generator becomes a crucial factor for virtually screening millions of structures. Crystal structure prediction (CSP) similarly relies on screening stable conformations, with recent models placing a greater emphasis on quality over quantity. These last two applications require modelling conformational flexibility rather than finding wells in the potential energy surface, as docked and crystallized conformations can fall outside these wells, carrying additional strain. The methods we present in this work can address any of these

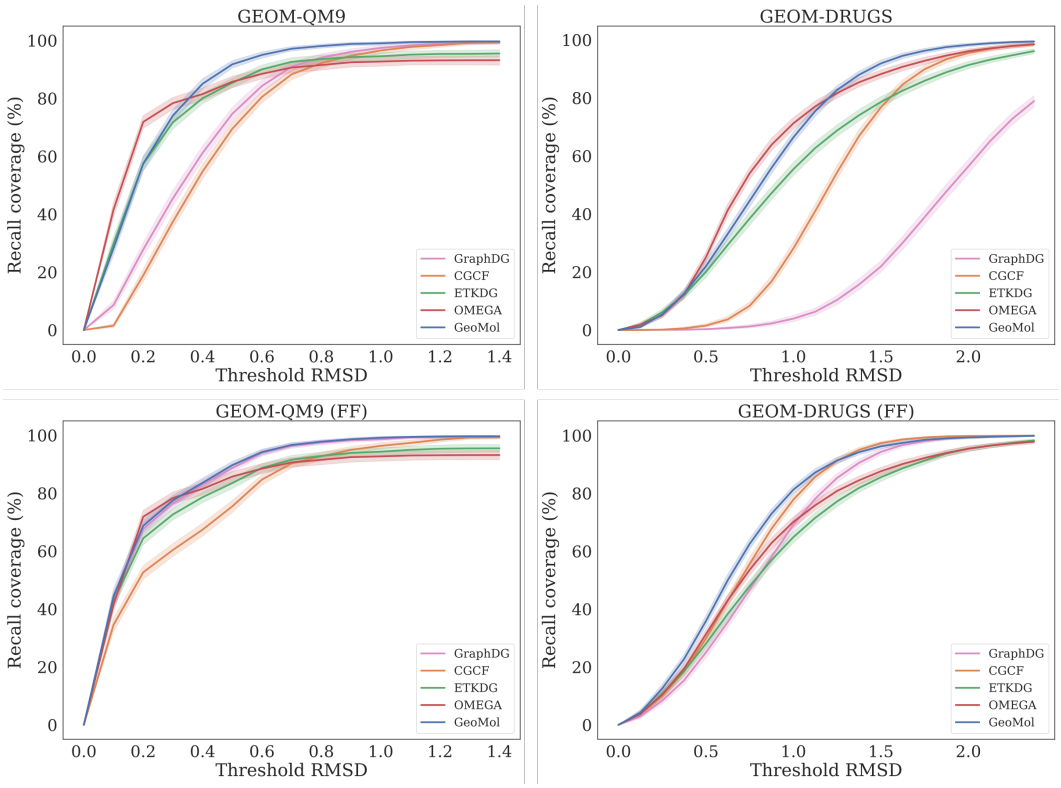

Figure 11: Recall coverage plotted against varying RMSD thresholds.

latter applications as well. We emphasize that the datasets on which GEOMOL (or any ML model) is trained guide the downstream usage, but cleverly finetuning GEOMOL on various downstream applications is an exciting future research direction.

O.2 Comment on evaluation metrics

Additionally, we note that while the evaluation metrics presented in this work are a sensible way to benchmark ML conformer generators against each other, they do not necessarily present a realistic evaluation of these algorithms in a production setting. This is why evaluating against the OMEGA software, which does not allow the user to request a specific number of conformers, is difficult. Further, for many applications, the generated conformers should be pruned and ranked (e.g., ranking conformers by energy and stable packing arrangements for CSP), and different conformer generation methods have varying methods to perform these post-processing steps. Because the scope of the presented work aims to distinguish our model against the many emerging ML-based conformer generators, we leave the exploration of such post-processing algorithms and strategies for future work. Furthermore, both the ETKDG and OMEGA algorithms are optimized to reproduce Protein Data Bank/Cambridge Structural Database conformations rather than replicate xTB low-energy conformations. In this sense, the evaluation presented here should not be viewed as a comprehensive argument for or against any of these methods.

O.3 QM9 results

Focusing on the QM9 results, we see both the power and drawbacks of the popular conformer generators such as OMEGA and ETKDG. Both methods can successfully generate diverse sets of conformers for the small molecules in the QM9 dataset, as shown by the high recall coverage values. However, both methods run into similar issues. That is, they both fail to generate the requested number of conformers for a few structures in the QM9 dataset, most of which have fused rings and multiple chiral centers specified. Since OMEGA, which constructs initial 3D structures from its

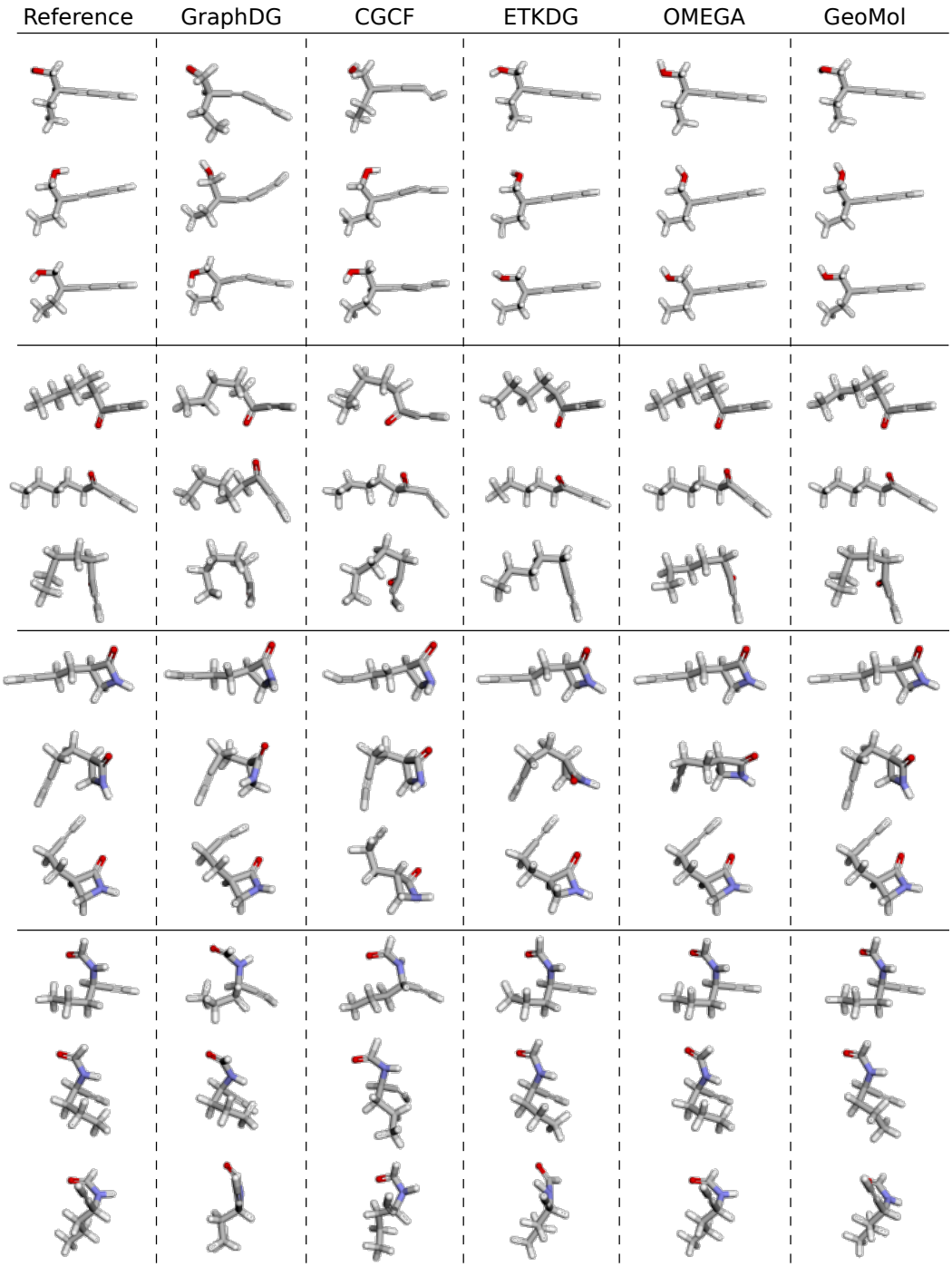

Figure 12: Additional examples of generated conformers for the GEOM-QM9 dataset.

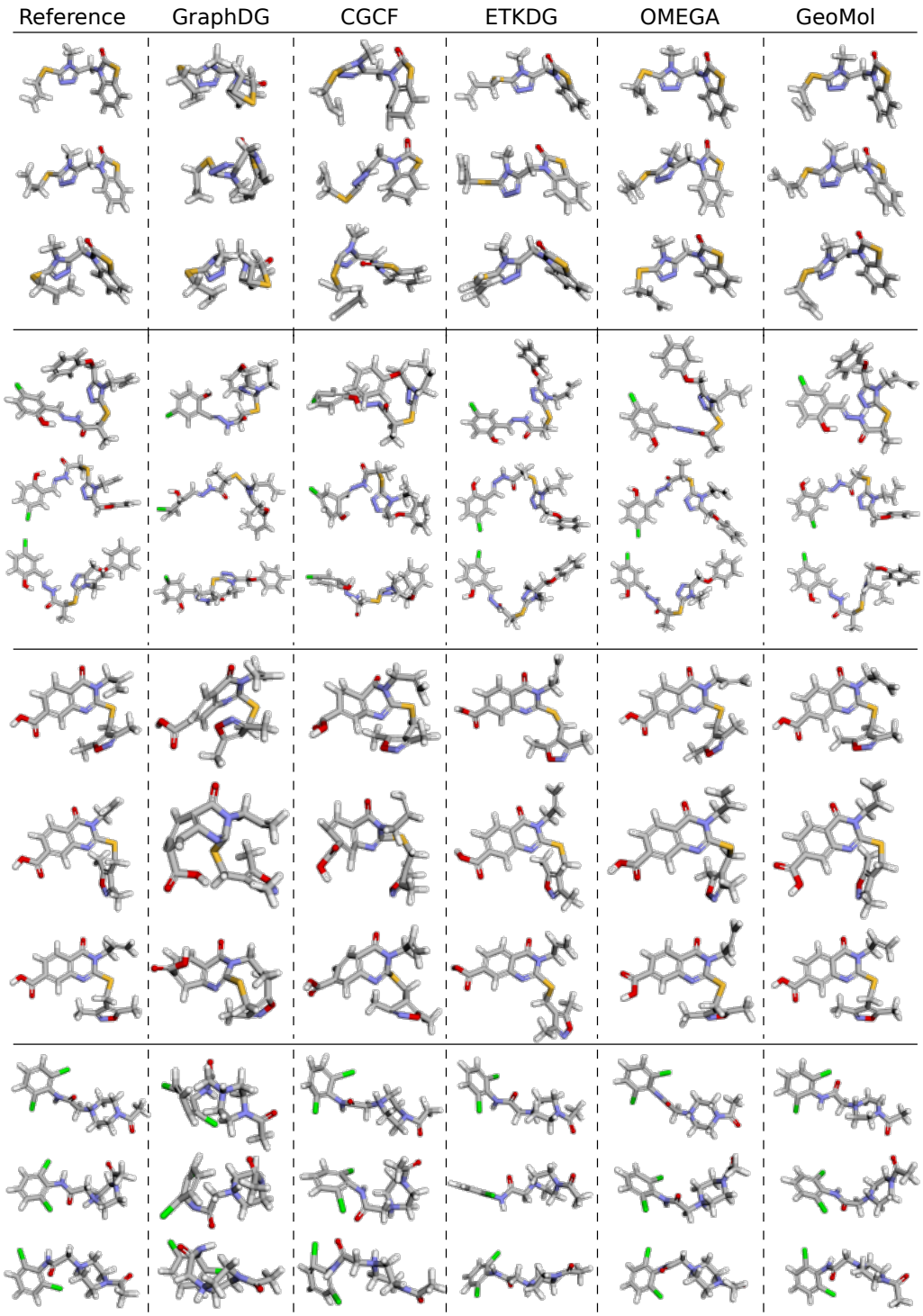

Figure 13: Additional examples of generated conformers for the GEOM-DRUGS dataset.

database of fragments, does not have many of these fragments readily available, both OMEGA and ETKDG likely run into the same issue: using DG-based methods to generate initial 3D conformers for molecules with multiple chiral centers is difficult because of the implicit chiral constraints that must be satisfied. Take for example the following SMILES, which represents a fairly simple fused ring structure: 'C#C[C@@]1(C)[C@H]2C[C@]1(C)C2'. ETKDG fails to embed a single conformer for this structure with repeated attempts. Removing the chiral centers specifications (i.e. inputting 'C#CC1(C)C2CC1(C)C2' as the SMILES) results in a successful structure generation, but many of the outputs have the incorrect chirality. While these can sometimes be corrected with energy-based minimization, the key issue of determining chiral centers exactly with DG-based methods remains. GEOMOL specifically tackles this problem such that all specified chiral centers are solved exactly (see section 2.3).

O.4 DRUGS results

The results for the DRUGS half of GEOM dataset suggest additional issues with traditional conformer generation methods. With these larger, drug-like molecules with many rotatable bonds, the search space of conformers grows exponentially. Here, it is instructive to separate the performance of ETKDG from OMEGA. ETKDG relies on randomly sampling the space of possible interatomic distances, which leads to poor coverage for conformers where this space is large. If the goal is to improve coverage, ETKDG could benefit from improved sampling strategies. OMEGA systematically varies all torsion angles in the structure to generate ensembles, but since this process can become prohibitively expensive, we must include an RMSD cutoff for many of these input molecules so as to return results in a reasonable period of time. Furthermore, not only does OMEGA uses a modified MMFF94 FF to generate its fragment library, it also uses a FF to score each potential conformer; its identified low-energy structures will be different than the xTB geometries we use as ground truths. This is a limitation with the OMEGA algorithm as currently constructed, and it could benefit from higher quality fragment structures and improved methods to score output conformers. This is not to understate the power of OMEGA as a conformer generation tool; it significantly outperforms the other baselines in our study and should be viewed as state-of-the-art in reference.

O.5 Runtime comparison

While the baselines ML methods have not benchmarked the speed of their algorithms, we believe this is a crucial evaluation of any conformer generator, especially if the intent is to use such a model for pose prediction and high throughput virtual screening. For each baseline, we request a fixed number of conformers and record the time necessary to output a single conformer, parallelizing each model over 40 CPU cores. We repeat this analysis three times for each method and plot the results in fig. 7. Importantly, we do not include the start up times for any of the methods (i.e. loading the neural network weights for the ML models). OMEGA, GraphDG, and GEOMOL show comparative performance, with ETKDG lagging slightly behind. Additionally, GEOMOL scales well for structures with an increasing number of rotatable bonds, especially compared to GraphDG and ETKDG. The CGCF model is orders of magnitude slower than the other baselines, which restricts its utility for high throughput applications of conformer generation.

O.6 Limitations

The most significant drawback of our model is its weakness in capturing long-range interactions, which can manifest itself in several ways, detailed in fig. 14. First, GEOMOL poorly models macrocycles. Because of the formulation we use to construct conformers, error accumulation sometimes leads to imperfect structures, as the ring smoothing algorithm breaks down for large cycles (top of fig. 14). We also qualitatively observe that large rings with very sparse examples in the training data can be predicted poorly by our model at test time. The middle of fig. 14 shows one such example of an eight-membered ring; here, additional training data should help, as there only exist a few eight-membered rings in the training set. Finally, the presence of steric clashes is an important issue with the current model. The bottom of fig. 14 shows one such example, being the most common error mode for GEOMOL. Since global conformer information during torsion predictions is only conveyed through the graph network representations, it is challenging to account for the position of atoms far away in graph connectivity, which can lead to overlapping atoms in the final prediction. This is especially troubling for interactions such as the pi-pi stacking depicted in the figure, which

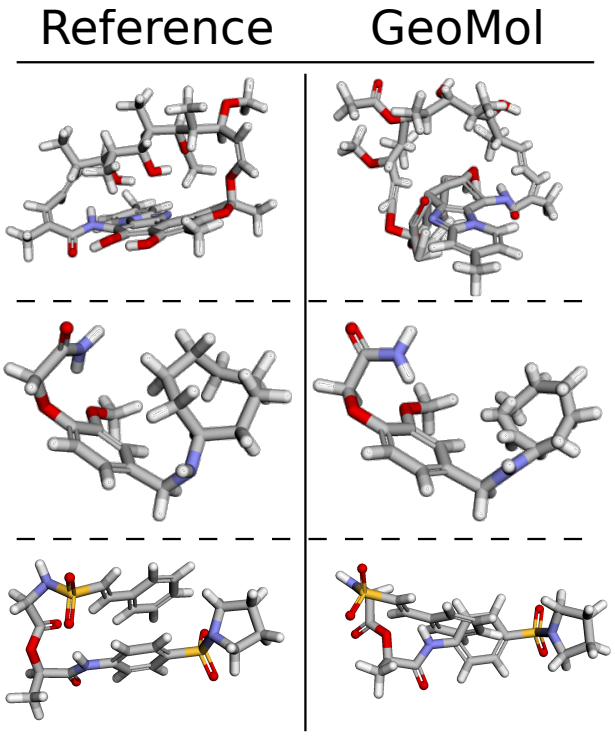

Figure 14: Common GEOMOL errors. Top: example of a macrocyle predicted poorly by GEOMOL. Middle: example of a larger ring with few training set occurrences. Bottom: example of a steric clash due to weak modeling of long-range interactions.

require moieties in the conformer with large 2D distances in the molecular graph to be predicted close to one another in 3D space. For this reason, extending GEOMOL with an improved method for long-range interactions is an exciting future direction.

O.7 Conclusions

In theory, GEOMOL should replicate the quality of conformers on which it was trained, which gives the model unique power and flexibility for downstream applications. For example, one could easily construct a smaller database of high-quality conformers optimized at a higher level of theory than semi-empirical (ex. coupled-cluster) and finetune GEOMOL. These higher-quality conformers could even include solvent effects in a range of solvents, which are often neglected in traditional conformer generation methods. Indeed, these downstream modifications could have significant impacts on a range of applications in cheminformatics and drug discovery.

Both the GraphDG and CGCF models, which represent important steps in ML-enabled conformer generation, are not quite ready to be used in a production setting without relying on a force field optimization. This is especially evident given the example structures presented in the main text and in appendix N. However, both models improve over traditional baselines at searching the space of interatomic distances and outputting diverse predictions, which is a credit to the search strategies employed by each model, especially CGCF. Further, the GraphDG model is competitive with GEOMOL for the QM9 dataset and should be viewed as a viable alternative to GEOMOL for small molecules with few heavy atoms.

As researchers continue to develop 3D ML-based conformer generators, the discussion between DG-based and non DG-based algorithms will carry on. Here, we chose to take the latter approach because we wanted to explicitly model important elements of conformer generation, and this strategy has been successful. In our experience, modeling these specific geometric elements leads to a significant improvement in the quality of the output structures. It is still unclear if such an approach can tackle

long-range interactions, which remains the major drawback of our model. We hope to answer this question in future work.