Torsional Diffusion for Molecular Conformer Generation

Bowen Jing,*[1] Gabriele Corso,*[1], Jeffrey Chang,[2] Regina Barzilay,[1] Tommi Jaakkola[1]
[1]CSAIL, Massachusetts Institute of Technology [2]Dept. of Physics, Harvard University

Abstract

Molecular conformer generation is a fundamental task in computational chemistry. Several machine learning approaches have been developed, but none have outperformed state-of-the-art cheminformatics methods. We propose *torsional diffusion*, a novel diffusion framework that operates on the space of torsion angles via a diffusion process on the hypertorus and an extrinsic-to-intrinsic score model. On a standard benchmark of drug-like molecules, torsional diffusion generates superior conformer ensembles compared to machine learning and cheminformatics methods in terms of both RMSD and chemical properties, and is orders of magnitude faster than previous diffusion-based models. Moreover, our model provides exact likelihoods, which we employ to build the first generalizable Boltzmann generator. Code is available at `https://github.com/gcorso/torsional-diffusion`.

1 Introduction

Many properties of a molecule are determined by the set of low-energy structures, called *conformers*, that it adopts in 3D space. Conformer generation is therefore a fundamental problem in computational chemistry [Hawkins, 2017] and an area of increasing attention in machine learning. Traditional approaches to conformer generation consist of metadynamics-based methods, which are accurate but slow [Pracht et al., 2020]; and cheminformatics-based methods, which are fast but less accurate [Hawkins et al., 2010, Riniker and Landrum, 2015]. Thus, there is growing interest in developing deep generative models to combine high accuracy with fast sampling.

Diffusion or score-based generative models [Ho et al., 2020, Song et al., 2021]—a promising class of generative models—have been applied to conformer generation under several different formulations. These have so far considered diffusion processes in *Euclidean* space, in which Gaussian noise is injected independently into every data coordinate—either pairwise distances in a distance matrix [Shi et al., 2021, Luo et al., 2021] or atomic coordinates in 3D [Xu et al., 2022]. However, these models require a large number of denoising steps and have so far failed to outperform the best cheminformatics methods.

We instead propose *torsional diffusion*, in which the diffusion process over conformers acts only on the torsion angles and leaves the other degrees of freedom fixed. This is possible and effective because the flexibility of a molecule, and thus the difficulty of conformer generation, lies largely in torsional degrees of freedom [Axelrod and Gómez-Bombarelli, 2022]; in particular, bond lengths and angles can already be determined quickly and accurately by standard cheminformatics methods. Leveraging this insight significantly reduces the dimensionality of the sample space; drug-like molecules[2] have, on average, $n = 44$ atoms, corresponding to a $3n$-dimensional Euclidean space, but only $m = 7.9$ torsion angles of rotatable bonds.

*Equal contribution. Correspondence to `{bjing, gcorso}@mit.edu`.

[2]As measured from the standard dataset GEOM-DRUGS [Axelrod and Gómez-Bombarelli, 2022]

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

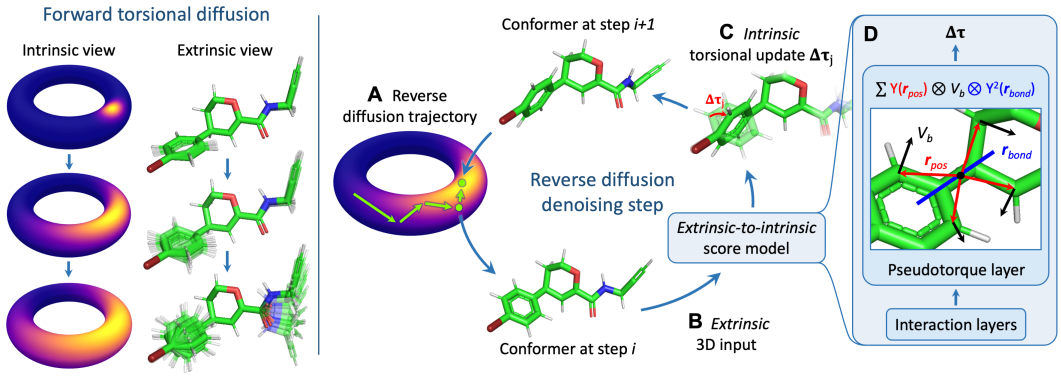

Figure 1: **Overview of torsional diffusion.** *Left*: Extrinsic and intrinsic views of torsional diffusion (only 2 dimensions/bonds shown). *Right*: In a step of reverse diffusion (**A**), the current conformer is provided as a 3D structure (**B**) to the score model, which predicts intrinsic torsional updates (**C**). The final layer of the score model is constructed to resemble a torque computation around each bond (**D**). Y refers to the spherical harmonics and V_b the learned atomic embeddings.

Torsion angle coordinates define not a Euclidean space, but rather an m-dimensional torus $\mathbb{T}^m$ (Figure 1, *left*). However, the dimensionality and distribution over the torus vary between molecules and even between different ways of defining the torsional space for the same molecule. To resolve these difficulties, we develop an *extrinsic-to-intrinsic* score model (Figure 1, *right*) that takes as input a 3D point cloud representation of the conformer in Euclidean space (extrinsic coordinates), and predicts as output a score on a torsional space *specific to that molecule* (intrinsic coordinates). To do so, we consider a *torsional score* for a bond as a geometric property of a 3D point cloud, and use $SE(3)$-equivariant networks to predict them directly for each bond.

Unlike prior work, our model provides exact likelihoods of generated conformers, enabling training with the ground-truth *energy* function rather than samples alone. This connects with the literature on *Boltzmann generators*—generative models which aim to sample the Boltzmann distribution of physical systems without expensive molecular dynamics or MCMC simulations [Noé et al., 2019, Köhler et al., 2021]. Thus, as a variation on the torsional diffusion framework, we develop *torsional Boltzmann generators* that can approximately sample the conditional Boltzmann distribution for unseen molecules. This starkly contrasts with existing Boltzmann generators, which are specific for the chemical system on which they are trained.

Our main contributions are:

- We formulate conformer generation in terms of diffusion modeling on the hypertorus—the first demonstration of non-Euclidean diffusion on complex datasets—and develop an extrinsic-to-intrinsic score model that satisfies the required symmetries: $SE(3)$ invariance, torsion definition invariance, and parity equivariance.
- We obtain state-of-the-art results on the GEOM-DRUGS dataset [Axelrod and Gómez-Bombarelli, 2022] and are the first method to consistently outperform the established commercial software OMEGA [Hawkins, 2017]. We do so using two orders of magnitude *fewer* denoising steps than GeoDiff [Xu et al., 2022], the best Euclidean diffusion approach.
- We propose torsional Boltzmann generators—the first Boltzmann generator based on diffusion models rather than normalizing flows and the first to be useful for a class of molecules rather than a specific system.

2 Background

Diffusion generative models Consider the data distribution as the starting distribution $p_0(\mathbf{x})$ of a *forward diffusion process* described by an Ito stochastic differential equation (SDE):

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}, \quad t \in (0, T) \tag{1}$$

where $\mathbf{w}$ is the Wiener process and $\mathbf{f}(\mathbf{x}, t), g(t)$ are chosen functions. With sufficiently large T, the distribution $p_T(\mathbf{x})$—the *prior*—approaches a simple Gaussian. Sampling from the prior and solving the *reverse diffusion*

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]\ dt + g(t)\ d\bar{\mathbf{w}} \tag{2}$$

yields samples from the data distribution $p_0(\mathbf{x})$ [Anderson, 1982, Song et al., 2021]. Diffusion, or score-based, generative models [Ho et al., 2020, Song et al., 2021] learn the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ of the diffused data with a neural network and generate data by approximately solving the reverse diffusion. The score of the diffused data also defines a *probability flow ODE*—a continuous normalizing flow that deterministically transforms the prior into the data distribution [Song et al., 2021]. We leverage the insight that, in many cases, this flow makes it possible to use diffusion models in place of normalizing flows and highlight one such case with the torsional Boltzmann generator.

Diffusion generative models have traditionally been used to model data on Euclidean spaces (such as images); however, De Bortoli et al. [2022] recently showed that the theoretical framework holds with relatively few modifications for data distributions on compact Riemannian manifolds. The hypertorus $\mathbb{T}^m$, which we use to define torsional diffusion, is a specific case of such a manifold.

Several methods [Salimans and Ho, 2022, Vahdat et al., 2021, Nichol and Dhariwal, 2021] have been proposed to improve and accelerate diffusion models in the domain of image generation. Among these, the most relevant to this work is *subspace diffusion* [Jing et al., 2022], in which the diffusion is progressively restricted to linear subspaces. Torsional diffusion can be viewed in a similar spirit, as it effectively restricts Euclidean diffusion to a nonlinear *manifold* given by fixing the non-torsional degrees of freedom.

Molecular conformer generation The *conformers* of a molecule are the set of its energetically favorable 3D structures, corresponding to local minima of the potential energy surface.[3] The gold standards for conformer generation are metadynamics-based methods such as CREST [Pracht et al., 2020], which explore the potential energy surface while filling in local minima [Hawkins, 2017]. However, these require an average of 90 core-hours per drug-like molecule [Axelrod and Gómez-Bombarelli, 2022] and are not considered suitable for high-throughput applications. Cheminformatics methods instead leverage approximations from chemical heuristics, rules, and databases for significantly faster generation [Lagorce et al., 2009, Cole et al., 2018, Miteva et al., 2010, Bolton et al., 2011, Li et al., 2007]; while these can readily model highly constrained degrees of freedom, they fail to capture the full energy landscape. The most well-regarded of such methods include the commercial software OMEGA [Hawkins et al., 2010] and the open-source RDKit ETKDG [Landrum et al., 2013, Riniker and Landrum, 2015].

A number of machine learning methods for conformer generation has been developed [Xu et al., 2021a,b, Shi et al., 2021, Luo et al., 2021], the most recent and advanced of which are GeoMol [Ganea et al., 2021] and GeoDiff [Xu et al., 2022]. GeoDiff is a Euclidean diffusion model that treats conformers as point clouds $\mathbf{x} \in \mathbb{R}^{3n}$ and learns an $SE(3)$ equivariant score. On the other hand, GeoMol employs a graph neural network that, in a single forward pass, predicts neighboring atomic coordinates and torsion angles from a stochastic seed.

Boltzmann generators An important problem in physics and chemistry is that of generating independent samples from a Boltzmann distribution $p(\mathbf{x}) \propto e^{-E(\mathbf{x})/kT}$ with known but unnormalized density.[4] Generative models with exact likelihoods, such as normalizing flows, can be trained to match such densities [Noé et al., 2019] and thus provide independent samples from an approximation of the target distribution. Such *Boltzmann generators* have shown high fidelity on small organic molecules [Köhler et al., 2021] and utility on systems as large as proteins [Noé et al., 2019]. However, a separate model has to be trained for every molecule, as the normalizing flows operate on intrinsic coordinates whose definitions are specific to that molecule. This limits the utility of existing Boltzmann generators for molecular screening applications.

[3] Conformers are typically considered up to an energy cutoff above the global minimum.

[4] This is related to but distinct from conformer generation, as conformers are the local minima of the Boltzmann distribution rather than independent samples.

3 Torsional Diffusion

Consider a molecule as a graph $G = (\mathcal{V}, \mathcal{E})$ with atoms $v \in \mathcal{V}$ and bonds $e \in \mathcal{E}$,[5] and denote the space of its possible conformers $\mathcal{C}_G$. A conformer $C \in \mathcal{C}_G$ can be specified in terms of its *intrinsic* (or internal) coordinates: local structures L consisting of bond lengths, bond angles, and cycle conformations; and torsion angles $\boldsymbol{\tau}$ consisting of dihedral angles around freely rotatable bonds (precise definitions in Appendix A). We consider a bond *freely rotatable* if severing the bond creates two connected components of G, each of which has at least two atoms.[6] Thus, torsion angles in cycles (or rings), which cannot be rotated independently, are considered part of the local structure L.

Conformer generation consists of learning probability distributions $p_G(L, \boldsymbol{\tau})$. However, the set of possible stable local structures L for a particular molecule is very constrained and can be accurately predicted by fast cheminformatics methods, such as RDKit ETKDG [Riniker and Landrum, 2015] (see Appendix F.1 for verification). Thus, we use RDKit to provide approximate samples from $p_G(L)$, and develop a diffusion-based generative model to learn distributions $p_G(\boldsymbol{\tau} \mid L)$ over torsion angles—conditioned on a given graph and local structure.

Our method is illustrated in Figure 1 and detailed as follows. Section 3.1 formulates diffusion modeling on the torus defined by torsion angles. Section 3.2 describes the torsional score framework, Section 3.3 the required symmetries, and Section 3.4 our score model architecture. Section 3.5 discusses likelihoods, and Section 3.6 how likelihoods can be used for energy-based training.

3.1 Diffusion modeling on $\mathbb{T}^m$

Since each torsion angle coordinate lies in $[0, 2\pi)$, the m torsion angles of a conformer define a hypertorus $\mathbb{T}^m$. To learn a generative model over this space, we apply the continuous score-based framework of Song et al. [2021], which holds with minor modifications for data distributions on compact Riemannian manifolds (such as $\mathbb{T}^m$) [De Bortoli et al., 2022]. Specifically, for Riemannian manifold M let $\mathbf{x} \in M$, let $\mathbf{w}$ be the Brownian motion on the manifold, and let the drift $\mathbf{f}(\mathbf{x}, t)$, score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, and score model output $\mathbf{s}(\mathbf{x}, t)$ be elements of the tangent space $T_{\mathbf{x}}M$. Then equation 2 remains valid—that is, discretizing and solving the reverse SDE on the manifold as a *geodesic random walk* starting with samples from $p_T(\mathbf{x})$ approximately recovers the original data distribution $p_0(\mathbf{x})$ [De Bortoli et al., 2022].

For the forward diffusion we use rescaled Brownian motion given by $\mathbf{f}(\mathbf{x}, t) = 0, g(t) = \sqrt{\frac{d}{dt}\sigma^2(t)}$ where $\sigma(t)$ is the noise scale. Specifically, we use an exponential diffusion $\sigma(t) = \sigma_{\text{min}}^{1-t}\sigma_{\text{max}}^{t}$ as in Song and Ermon [2019], with $\sigma_{\text{min}} = 0.01\pi$, $\sigma_{\text{max}} = \pi, t \in (0, 1)$. Due to the compactness of the manifold, however, the prior $p_T(\mathbf{x})$ is no longer a Gaussian, but a *uniform* distribution over M.

Training the score model with denoising score matching requires a procedure to sample from the perturbation kernel $p_{t|0}(\mathbf{x}' \mid \mathbf{x})$ of the forward diffusion and compute its score. We view the torus $\mathbb{T}^m \cong [0, 2\pi)^m$ as the quotient space $\mathbb{R}^m/2\pi\mathbb{Z}^m$ with equivalence relations $(\tau_1, \dots \tau_m) \sim (\tau_1 + 2\pi, \dots, \tau_m) \dots \sim (\tau_1, \dots \tau_m + 2\pi)$. Hence, the perturbation kernel for rescaled Brownian motion on $\mathbb{T}^m$ is the *wrapped normal distribution* on $\mathbb{R}^m$; that is, for any $\boldsymbol{\tau}, \boldsymbol{\tau}' \in [0, 2\pi)^m$, we have

$$p_{t|0}(\boldsymbol{\tau}' \mid \boldsymbol{\tau}) \propto \sum_{\mathbf{d} \in \mathbb{Z}^m} \exp\left(-\frac{||\boldsymbol{\tau} - \boldsymbol{\tau}' + 2\pi\mathbf{d}||^2}{2\sigma^2(t)}\right) \tag{3}$$

where $\sigma(t)$ is the noise scale of the perturbation kernel $p_{t|0}$. We thus sample from the perturbation kernel by sampling from the corresponding unwrapped isotropic normal and taking elementwise $\bmod 2\pi$. The scores of the kernel are pre-computed using a numerical approximation. During training, we sample times t at uniform and minimize the denoising score matching loss

$$J_{\text{DSM}}(\theta) = \mathbb{E}_t \left[\lambda(t)\mathbb{E}_{\boldsymbol{\tau}_0 \sim p_0, \boldsymbol{\tau}_t \sim p_{t|0}(\cdot|\boldsymbol{\tau}_0)} \left[||\mathbf{s}(\boldsymbol{\tau}_t, t) - \nabla_{\boldsymbol{\tau}_t} \log p_{t|0}(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_0)||^2\right]\right] \tag{4}$$

where the weight factors $\lambda(t) = 1/\mathbb{E}_{\boldsymbol{\tau} \sim p_{t|0}(\cdot|0)} \left[||\nabla_{\boldsymbol{\tau}} \log p_{t|0}(\boldsymbol{\tau} \mid \mathbf{0})||^2\right]$ are also precomputed. As the tangent space $T_{\boldsymbol{\tau}}\mathbb{T}^m$ is just $\mathbb{R}^m$, all the operations in the loss computation are the familiar ones.

[5] Chirality and other forms of stereoisomerism are discussed in Appendix F.3.

[6] Notably, this counts double bonds as rotatable. See Appendix F.3 for further discussion.

For inference, we first sample from a uniform prior over the torus. We then discretize and solve the reverse diffusion with a geodesic random walk; however, since the exponential map on the torus (viewed as a quotient space) is just $\exp_{\boldsymbol{\tau}}(\boldsymbol{\delta}) = \boldsymbol{\tau} + \boldsymbol{\delta} \mod 2\pi$, the geodesic random walk is equivalent to the wrapping of the random walk on $\mathbb{R}^m$.

3.2 Torsional score framework

While we have defined the diffusion process over intrinsic coordinates, learning a score model $\mathbf{s}(\boldsymbol{\tau}, t)$ directly over intrinsic coordinates is potentially problematic for several reasons. First, the dimensionality m of the torsional space depends on the molecular graph G. Second, the mapping from torsional space to physically distinct conformers depends on G and local structures L, but it is unclear how to best provide these to a model over $\mathbb{T}^m$. Third, there is no canonical choice of independent intrinsic coordinates $(L, \boldsymbol{\tau})$; in particular, the torsion angle at a rotatable bond can be defined as any of the dihedral angles at that bond, depending on an arbitrary choice of reference neighbors (Figure 2 and Appendix A). Thus, even with fixed G and L, the mapping from $\mathbb{T}^m$ to conformers is ill-defined. This posed a significant challenge to prior works using intrinsic coordinates [Ganea et al., 2021].

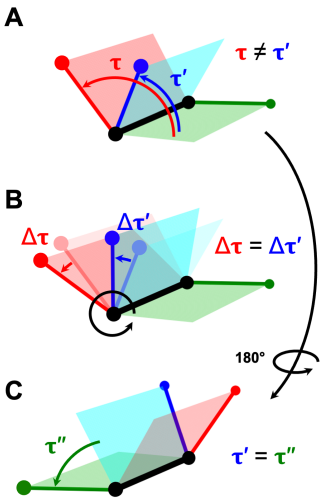

Figure 2: **A**: The torsion τ around a bond depends on a choice of neighbors. **B**: The *change* $\Delta\tau$ caused by a relative rotation is the same for all choices. **C**: The sign of $\Delta\tau$ is unambiguous because given the same neighbors, τ does not depend on bond direction.

To circumvent these difficulties, we instead consider a conformer $C \in \mathcal{C}_G$ in terms of its *extrinsic* (or Cartesian) coordinates—that is, as a point cloud in 3D space, defined up to global roto-translation: $\mathcal{C}_G \cong \mathbb{R}^{3n}/SE(3)$. Then, we construct the score model $\mathbf{s}_G(C, t)$ as a function over $\mathcal{C}_G$ rather than $\mathbb{T}^m$. The outputs remain in the tangent space of $\mathbb{T}^m$, which is just $\mathbb{R}^m$. Such a score model is simply an $SE(3)$-*invariant* model over point clouds in 3D space $\mathbf{s}_G : \mathbb{R}^{3n} \times [0, T] \mapsto \mathbb{R}^m$ conditioned on G. Thus, we have reduced the problem of learning a score on the torus, conditioned on the molecular graph and local structure, to the much more familiar problem of predicting $SE(3)$-invariant scalar quantities—one for each bond—from a 3D conformer.

It may appear that we still need to choose a definition of each torsion angle τ_i so that we can sample from $p_{t|0}(\cdot \mid \boldsymbol{\tau})$ during training and solve the reverse SDE over $\boldsymbol{\tau}$ during inference. However, we leverage the following insight: given *fixed local structures*, the action on C of changing a single torsion angle τ_i by some $\Delta\tau_i$ can be applied without choosing a definition (Figure 2). Geometrically, this action is a (signed) relative rotation of the atoms on opposite sides of the bond and can be applied directly to the atomic coordinates in 3D. The geometric intuition can be stated as follows (proven in Appendix B and discussed further in Appendix F.2).

Proposition 1. *Let (b_i, c_i) be a rotatable bond, let $\mathbf{x}_{\mathcal{V}(b_i)}$ be the positions of atoms on the b_i side of the molecule, and let $R(\boldsymbol{\theta}, x_{c_i}) \in SE(3)$ be the rotation by Euler vector $\boldsymbol{\theta}$ about x_{c_i}. Then for $C, C' \in \mathcal{C}_G$, if τ_i is any definition of the torsion angle around bond (b_i, c_i),*

$$\begin{aligned} \tau_i(C') &= \tau_i(C) + \theta \\ \tau_j(C') &= \tau_j(C) \quad \forall j \neq i \end{aligned} \qquad \textit{if} \qquad \exists \mathbf{x} \in C, \mathbf{x}' \in C'. \quad \begin{aligned} \mathbf{x}'_{\mathcal{V}(b_i)} &= \mathbf{x}_{\mathcal{V}(b_i)} \\ \mathbf{x}'_{\mathcal{V}(c_i)} &= R\left(\theta\, \hat{\mathbf{r}}_{b_i c_i}, x_{c_i}\right) \mathbf{x}_{\mathcal{V}(c_i)} \end{aligned} \tag{5}$$

where $\hat{\mathbf{r}}_{b_i c_i} = (x_{c_i} - x_{b_i})/||x_{c_i} - x_{b_i}||$.

To apply a torsion update $\Delta\boldsymbol{\tau} = (\Delta\tau_1, \ldots \Delta\tau_m)$ involving all bonds, we apply $\Delta\tau_i$ sequentially in any order. Then, since training and inference only make use of torsion updates $\Delta\boldsymbol{\tau}$, we work solely in terms of 3D point clouds and updates applied to them. To draw local structures L from RDKit, we draw full 3D conformers $C \in \mathcal{C}_G$ and then randomize all torsion angles to sample uniformly over $\mathbb{T}^m$. To solve the reverse SDE, we repeatedly predict torsion updates directly from, and apply them directly to, the 3D point cloud. Therefore, since our method never requires a choice of reference neighbors for any τ_i, it is manifestly invariant to such a choice. These procedures are detailed in Appendix C.

3.3 Parity equivariance

The torsional score framework presented thus far requires an $SE(3)$-invariant model. However, an additional symmetry requirement arises from the fact that the underlying physical energy is invariant, or extremely nearly so, under *parity inversion* [Quack, 2002]. Thus our learned density should respect $p(C) = p(-C)$ where $-C = \{-\mathbf{x} \mid \mathbf{x} \in C\}$. In terms of the conditional distribution over torsion angles, we require $p(\boldsymbol{\tau}(C) \mid L(C)) = p(\boldsymbol{\tau}(-C) \mid L(-C))$. Then,

Proposition 2. *If $p(\boldsymbol{\tau}(C) \mid L(C)) = p(\boldsymbol{\tau}(-C) \mid L(-C))$, then for all diffusion times t,*

$$\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}(C) \mid L(C)) = -\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}(-C) \mid L(-C)) \tag{6}$$

Because the score model seeks to learn $\mathbf{s}_G(C, t) = \nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}(C) \mid L(C))$, we must have $\mathbf{s}_G(C, t) = -\mathbf{s}_G(-C, t)$. Thus, the score model must be *invariant* under $SE(3)$ but *equivariant* (change sign) under parity inversion of the input point cloud— i.e. it must output a set of *pseudoscalars* in $\mathbb{R}^m$.

3.4 Score network architecture

Based on sections 3.2 and 3.3, the desiderata for the score model are:

Predict a pseudoscalar $\delta\tau_i := \partial \log p / \partial \tau_i \in \mathbb{R}$ that is $SE(3)$-invariant and parity equivariant for every rotatable bond in a 3D point cloud representation of a conformer.

While there exist several GNN architectures which are $SE(3)$-equivariant [Jing et al., 2021, Satorras et al., 2021], their $SE(3)$-invariant outputs are also parity invariant and, therefore, cannot satisfy the desired symmetry. Instead, we leverage the ability of equivariant networks based on tensor products [Thomas et al., 2018, Geiger et al., 2022] to produce pseudoscalar outputs.

Our architecture, detailed in Appendix D, consists of an embedding layer, a series of atomic convolution layers, and a final bond convolution layer. The first two closely follow the architecture of Tensor Field Networks [Thomas et al., 2018], and produce learned feature vectors for each atom. The final bond convolution layer constructs tensor product filters spatially centered on every rotatable bond and aggregates messages from neighboring atom features. We extract the pseudoscalar outputs of this filter to produce a single real-valued pseudoscalar prediction $\delta\tau_i$ for each rotatable bond.

Naively, the bond convolution layer could be constructed the same way as the atomic convolution layers, i.e., with spherical harmonic filters. However, to supply information about the orientation of the bond about which the torsion occurs, we construct a filter from the product of the spherical harmonics with a representation of the bond (Figure 1D). Because the convolution conceptually resembles computing the torque, we call this final layer the *pseudotorque* layer.

3.5 Likelihood

By using the probability flow ODE, we can compute the likelihood of any sample $\boldsymbol{\tau}$ as follows [Song et al., 2021, De Bortoli et al., 2022]:

$$\log p_0(\boldsymbol{\tau}_0) = \log p_T(\boldsymbol{\tau}_T) - \frac{1}{2} \int_0^T g^2(t)\ \nabla_{\boldsymbol{\tau}} \cdot \mathbf{s}_G(\boldsymbol{\tau}_t, t)\ dt \tag{7}$$

In Song et al. [2021], the divergence term is approximated via Hutchinson's method [Hutchinson, 1989], which gives an unbiased estimate of $\log p_0(\boldsymbol{\tau})$. However, this gives a *biased* estimate of $p_0(\boldsymbol{\tau})$, which is unsuitable for our applications. Thus, we compute the divergence term directly, which is feasible here (unlike in Euclidean diffusion) due to the reduced dimensionality of the torsional space.

The above likelihood is in *torsional* space $p_G(\boldsymbol{\tau} \mid L), \boldsymbol{\tau} \in \mathbb{T}^m$, but to enable compatibility with the Boltzmann measure $e^{-E(\mathbf{x})/kT}$, it is desirable to interconvert this with a likelihood in *Euclidean* space $p(\mathbf{x} \mid L), \mathbf{x} \in \mathbb{R}^{3n}$. A factor is necessary to convert between the volume element in torsional space and in Euclidean space (full derivation in Appendix B):

Proposition 3. *Let $\mathbf{x} \in C(\boldsymbol{\tau}, L)$ be a centered[7] conformer in Euclidean space. Then,*

$$p_G(\mathbf{x} \mid L) = \frac{p_G(\boldsymbol{\tau} \mid L)}{8\pi^2 \sqrt{\det g}} \quad \text{where } g_{\alpha\beta} = \sum_{k=1}^{n} J_\alpha^{(k)} \cdot J_\beta^{(k)} \tag{8}$$

[7] Additional formalism is needed for translations, but it is independent of the conformer and can be ignored.

where the indices α, β are integers between 1 and $m+3$. For $1 \leq \alpha \leq m$, $J_\alpha^{(k)}$ is defined as

$$J_i^{(k)} = \tilde{J}_i^{(k)} - \frac{1}{n}\sum_{\ell=1}^{n} \tilde{J}_i^{(\ell)} \quad \text{with} \quad \tilde{J}_i^{(\ell)} = \begin{cases} 0 & \ell \in \mathcal{V}(b_i), \\ \frac{\mathbf{x}_{b_i} - \mathbf{x}_{c_i}}{||\mathbf{x}_{b_i} - \mathbf{x}_{c_i}||} \times (\mathbf{x}_\ell - \mathbf{x}_{c_i}), & \ell \in \mathcal{V}(c_i), \end{cases} \tag{9}$$

and for $\alpha \in \{m+1, m+2, m+3\}$ as

$$J_{m+1}^{(k)} = \mathbf{x}_k \times \hat{x}, \qquad J_{m+2}^{(k)} = \mathbf{x}_k \times \hat{y}, \qquad J_{m+3}^{(k)} = \mathbf{x}_k \times \hat{z}, \tag{10}$$

where (b_i, c_i) is the freely rotatable bond for torsion angle i, $\mathcal{V}(b_i)$ is the set of all nodes on the same side of the bond as b_i, and $\hat{x}, \hat{y}, \hat{z}$ are the unit vectors in the respective directions.

3.6 Energy-based training

By computing likelihoods, we can train torsional diffusion models to match the Boltzmann distribution over torsion angles using the energy function. At a high level, we minimize the usual score matching loss, but with simulated samples from the Boltzmann distribution rather than data samples. The procedure therefore consists of two stages: resampling and score matching, which are tightly coupled during training (Algorithm 1). In the *resampling* stage, we use the model as an importance sampler for the Boltzmann distribution, where Proposition 3 is used to compute the (unnormalized) torsional Boltzmann density $\tilde{p}_G(\boldsymbol{\tau} \mid L)$. In the *score-matching* stage, the importance weights are used to approximate the denoising score-matching loss with expectations taken over $\tilde{p}_G(\boldsymbol{\tau} \mid L)$. As the model learns the score, it improves as an importance sampler.

Algorithm 1: Energy-based training epoch

Input: Boltzmann density $\tilde{p}$, training pairs $\{(G_i, L_i)\}_i$, torsional diffusion model q
for *each* (G_i, L_i) **do**
  Sample $\boldsymbol{\tau}_1, \ldots \boldsymbol{\tau}_K \sim q_{G_i}(\boldsymbol{\tau} \mid L_i)$;
  for $k \leftarrow 1$ **to** K **do**
    $\tilde{w}_k = \tilde{p}_{G_i}(\boldsymbol{\tau}_k \mid L_i)/q_{G_i}(\boldsymbol{\tau}_k \mid L_i)$;
  Approximate J_{DSM} for $p_0 \propto \tilde{p}$ using $\{(\tilde{w}_i, \boldsymbol{\tau}_i)\}_i$;
  Minimize J_{DSM};

This training procedure differs substantially from that of existing Boltzmann generators, which are trained as flows with a loss that directly depends on the model density. In contrast, we *train* the model as a score-based model, but *use* it as a flow—both during training and inference—to generate samples. The model density is needed only to reweight the samples to approximate the target density. Since in principle the model used for resampling does not need to be the same as the model being trained,[8] we can use very few steps (a shallow flow) during resampling to accelerate training, and then increase the number of steps (a deeper flow) for better approximations during inference—an option unavailable to existing Boltzmann generators.

4 Experiments

We evaluate torsional diffusion by comparing the generated and ground-truth conformers in terms of ensemble RMSD (Section 4.3) and properties (Section 4.4). Section 4.1 first discusses a preprocessing procedure required to train a conditional model $p_G(\boldsymbol{\tau} \mid L)$. Section 4.5 concludes with torsional Boltzmann generators. See Appendix H for additional results, including ablation experiments.

4.1 Conformer matching

In focusing on $p_G(\boldsymbol{\tau} \mid L)$, we have assumed that we can sample local structures $L \sim p_G(L)$ with RDKit. While this assumption is very good in terms of RMSD, the RDKit marginal $\hat{p}_G(L)$ is only an approximation of the ground truth $p_G(L)$. Thus, if we train on the denoising score-matching loss with ground truth conformers—i.e., conditioned on ground truth local structures—there will be a distributional shift at test time, where only approximate local structures from $\hat{p}_G(L)$ are available. We found that this shift significantly hurts performance.

We thus introduce a preprocessing procedure called *conformer matching*. In brief, for the *training* split only, we substitute each ground truth conformer C with a synthetic conformer $\hat{C}$ with local structures $\hat{L} \sim \hat{p}_G(L)$ and made as similar as possible to C. That is, we use RDKit to generate

[8] For example, if the resampler were perfect, the procedure would reduce to normal denoising score matching.

Table 1: Quality of generated conformer ensembles for the GEOM-DRUGS test set in terms of Coverage (%) and Average Minimum RMSD (Å). We compute Coverage with a threshold of $\delta = 0.75$ Å to better distinguish top methods. Note that this is different from most prior works, which used $\delta = 1.25$ Å.

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
RDKit ETKDG	38.4	28.6	1.058	1.002	40.9	30.8	0.995	0.895
OMEGA	53.4	54.6	0.841	0.762	40.5	33.3	0.946	0.854
GeoMol	44.6	41.4	0.875	0.834	43.0	36.4	0.928	0.841
GeoDiff	42.1	37.8	0.835	0.809	24.9	14.5	1.136	1.090
Torsional Diffusion	**72.7**	**80.0**	**0.582**	**0.565**	**55.2**	**56.9**	**0.778**	**0.729**

$\hat{L}$ and change torsion angles $\hat{\boldsymbol{\tau}}$ to minimize $\mathrm{RMSD}(C, \hat{C})$. Naively, we could sample $\hat{L} \sim \hat{p}_G(L)$ independently for each conformer, but this eliminates any possible dependence between L and $\boldsymbol{\tau}$ that could serve as training signal. Instead, we view the distributional shift as a domain adaptation problem that can be solved by optimally aligning $p_G(L)$ and $\hat{p}_G(L)$. See Appendix E for details.

4.2 Experimental setup

Dataset We evaluate on the GEOM dataset [Axelrod and Gómez-Bombarelli, 2022], which provides gold-standard conformer ensembles generated with metadynamics in CREST [Pracht et al., 2020]. We focus on GEOM-DRUGS—the largest and most pharmaceutically relevant part of the dataset—consisting of 304k drug-like molecules (average 44 atoms). To test the capacity to extrapolate to the largest molecules, we also collect from GEOM-MoleculeNet all species with more than 100 atoms into a dataset we call GEOM-XL and use it to evaluate models trained on DRUGS. Finally, we train and evaluate models on GEOM-QM9, a more established dataset but with significantly smaller molecules (average 11 atoms). Results for GEOM-XL and GEOM-QM9 are in Appendix H.

Evaluation We use the train/val/test splits from Ganea et al. [2021] and use the same metrics to compare the generated and ground truth conformer ensembles: Average Minimum RMSD (AMR) and Coverage. These metrics are reported both for Recall (R)—which measures how well the generated ensemble covers the ground-truth ensemble—and Precision (P)—which measures the accuracy of the generated conformers. See Appendix G for exact definitions and further details. Following the literature, we generate $2K$ conformers for a molecule with K ground truth conformers.

Baselines We compare with the strongest existing methods from Section 2. Among cheminformatics methods, we evaluate RDKit ETKDG [Riniker and Landrum, 2015], the most established open-source package, and OMEGA [Hawkins et al., 2010, Hawkins and Nicholls, 2012], a commercial software in continuous development. Among machine learning methods, we evaluate GeoMol [Ganea et al., 2021] and GeoDiff [Xu et al., 2022], which have outperformed all previous models on the evaluation metrics. Note that GeoDiff originally used a small subset of the DRUGS dataset, so we retrained it using the splits from Ganea et al. [2021].

4.3 Ensemble RMSD

Torsional diffusion significantly outperforms all previous methods on GEOM-DRUGS (Table 1 and Figure 3), reducing by 30% the average minimum recall RMSD and by 16% the precision RMSD relative to the previous state-of-the-art method. Torsional diffusion is also the first ML method to consistently generate better ensembles than OMEGA. As OMEGA is a well-established product used in industry, this represents an essential step towards establishing the utility of conformer generation with machine learning.

Torsional diffusion offers specific advantages over both GeoDiff and GeoMol, the most advanced prior machine learning methods. GeoDiff, a Euclidean diffusion model, requires 5000 denoising steps to obtain the results shown, whereas our model—thanks to the reduced degrees of freedom—requires only 20 steps. In fact, our model outperforms GeoDiff with as few as 5 denoising steps. As seen in Table 2, this translates to enormous runtime improvements.

Figure 3: Mean coverage for recall (*left*) and precision (*right*) when varying the threshold value δ on GEOM-DRUGS.

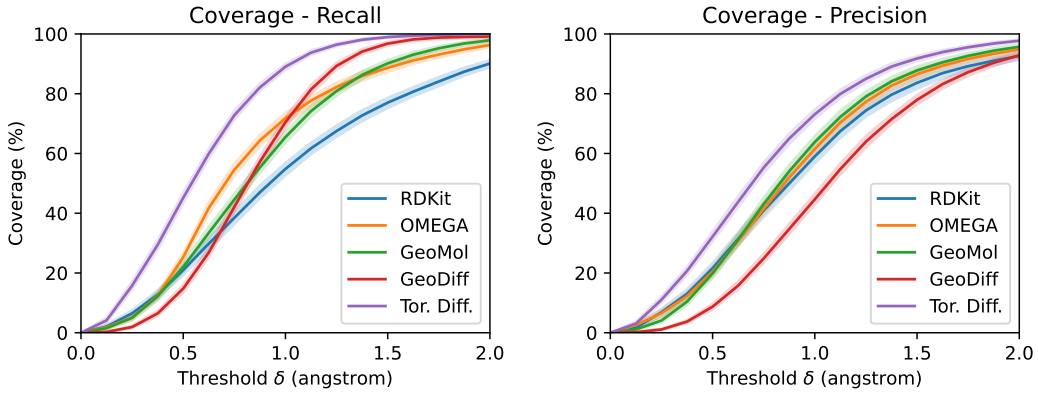

Table 2: Median AMR and runtime (core-secs per conformer) of machine learning methods, evaluated on CPU for comparison with RDKit.

Method	Steps	AMR-R	AMR-P	Runtime
RDKit	-	1.002	0.895	**0.10**
GeoMol	-	0.834	0.841	0.18
GeoDiff	5000	0.809	1.090	305
Torsional Diffusion	5	0.685	0.963	1.76
	10	0.580	0.791	2.82
	20	**0.565**	**0.729**	4.90

Table 3: Median absolute error of generated v.s. ground truth ensemble properties. E, $\Delta\epsilon$, $E_{\min}$ in kcal/mol, μ in debye.

Method	E	μ	$\Delta\epsilon$	$E_{\min}$
RDKit	0.81	0.52	0.75	1.16
OMEGA	0.68	0.66	0.68	0.69
GeoMol	0.42	**0.34**	0.59	0.40
GeoDiff	0.31	0.35	0.89	0.39
Tor. Diff.	**0.22**	0.35	**0.54**	**0.13**

Compared to torsional diffusion, GeoMol similarly makes use of intrinsic coordinates. However, since GeoMol can only access the molecular graph, it is less suited for reasoning about relationships that emerge only in a spatial embedding, especially between regions of the molecule that are distant on the graph. Our extrinsic-to-intrinsic score framework—which gives direct access to spatial relationships—addresses precisely this issue. The empirical advantages are most evident for the large molecules in GEOM-XL, on which GeoMol fails to improve consistently over RDKit (Appendix H). On the other hand, because GeoMol requires only a single-forward pass, it retains the advantage of faster runtime compared to diffusion-based methods.

4.4 Ensemble properties

While RMSD gives a *geometric* way to evaluate ensemble quality, we also consider the *chemical* similarity between generated and ground truth ensembles. For a random 100-molecule subset of DRUGS, we generate $\min(2K, 32)$ conformers per molecule, relax the conformers with GFN2-xTB [Bannwarth et al., 2019],[9] and compare the Boltzmann-weighted properties of the generated and ground truth ensembles. Specifically, the following properties are computed with xTB [Bannwarth et al., 2019]: energy E, dipole moment μ, HOMO-LUMO gap $\Delta\epsilon$, and the minimum energy $E_{\min}$. The median errors for torsional diffusion and the baselines are shown in Table 4. Our method produces the most chemically accurate ensembles, especially in terms of energy. In particular, we significantly improve over GeoMol and GeoDiff in finding the lowest-energy conformers that are only (on median) 0.13 kcal/mol higher in energy than the global minimum.

[9]Results without relaxation (which are less chemically meaningful) are in Appendix H.

4.5 Torsional Boltzmann generator

Finally, we evaluate how well a torsional Boltzmann generator trained with MMFF [Halgren, 1996] energies can sample the corresponding Boltzmann density over torsion angles. We train and test on GEOM-DRUGS molecules with 3–7 rotatable bonds and use the local structures of the first ground-truth conformers. For the baselines, we implement annealed importance samplers (AIS) [Neal, 2001] with Metropolis-Hastings steps over the torsional space and tune the variance of the transition kernels.

Table 4: Effective sample size (out of 32) given by importance sampling weights over the torsional Boltzmann density.

		Temp. (K)		
Method	Steps	1000	500	300
Uniform	–	1.71	1.21	1.02
AIS	5	2.20	1.36	1.18
	20	3.12	1.76	1.30
	100	6.72	3.12	2.06
Torsional BG	5	7.28	3.60	3.04
	20	**11.42**	**6.42**	**4.68**

Table 4 shows the quality of the samplers in terms of the *effective sample size* (ESS) given by the weights of 32 samples for each test molecule, which measures the α-divergence (with $\alpha = 2$) between the model and Boltzmann distributions [Midgley et al., 2021]. Our method significantly outperforms the AIS baseline, and improves with increased step size despite being trained with only a 5-step resampler. Note that, since these evaluations are done on *unseen* molecules, they are beyond the capabilities of existing Boltzmann generators.

5 Conclusion

We presented *torsional diffusion*, a method for generating molecular conformers based on a diffusion process restricted to the most flexible degrees of freedom. Torsional diffusion is the first machine learning model to significantly outperform standard cheminformatics methods and is orders of magnitude faster than previous Euclidean diffusion models. Using the exact likelihoods provided by our model, we also train the first system-agnostic Boltzmann generator.

There are several exciting avenues for future work. A natural extension is to relax the rigid local structure assumption by developing an efficient diffusion-based model over the full space of intrinsic coordinates while still incorporating chemical constraints. Moreover, torsional diffusion—or similar ideas—could be applicable to larger molecular systems, for which fast, parsimonious models of structural flexibility could benefit applications such as drug discovery and protein design.

Acknowledgments

We pay tribute to Octavian-Eugen Ganea (1987-2022), dear colleague, mentor, and friend without whom this work would have never been possible.

We thank Hannes Stärk, Wenxian Shi, Xiang Fu, Felix Faltings, Jason Yim, Adam Fisch, Alex Wu, Jeremy Wohlwend, Peter Mikhael, and Saro Passaro for helpful feedback and discussions. We thank Lagnajit Pattanaik, Minkai Xu, and Simon Axelrod for their advice and support when working with, respectively, GeoMol, GeoDiff and the GEOM dataset. This work was supported by the Machine Learning for Pharmaceutical Discovery and Synthesis (MLPDS) consortium, the Abdul Latif Jameel Clinic for Machine Learning in Health, the DTRA Discovery of Medical Countermeasures Against New and Emerging (DOMANE) threats program, the DARPA Accelerated Molecular Discovery program and the Sanofi Computational Antibody Design grant. We acknowledge support from the Department of Energy Computational Science Graduate Fellowship (BJ), the Robert Shillman Fellowship (GC), and the NSF Graduate Research Fellowship (JC).

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 1982.

Simon Axelrod and Rafael Gómez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 2022.

Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. Gfn2-xtb—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of chemical theory and computation*, 2019.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 2022.

Evan E Bolton, Sunghwan Kim, and Stephen H Bryant. Pubchem3d: conformer generation. *Journal of cheminformatics*, 2011.

Sean M Carroll. *Spacetime and geometry*. Cambridge University Press, 2019.

Gregory S Chirikjian. *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*, volume 2. Springer Science & Business Media, 2011.

Jason C Cole, Oliver Korb, Patrick McCabe, Murray G Read, and Robin Taylor. Knowledge-based conformer generation using the cambridge structural database. *Journal of Chemical Information and Modeling*, 2018.

David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 2016.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modeling. *arXiv preprint*, 2022.

Edward M Driggers, Stephen P Hale, Jinbo Lee, and Nicholas K Terrett. The exploration of macrocycles for drug discovery—an underexploited structural class. *Nature Reviews Drug Discovery*, 2008.

Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 2021.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022. URL `https://doi.org/10.5281/zenodo.6459381`.

Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 1996.

Paul CD Hawkins. Conformation generation: the state of the art. *Journal of Chemical Information and Modeling*, 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*, 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*, 2010.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 1989.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. *arXiv preprint arXiv:2205.01490*, 2022.

Leslie Kish. *Survey sampling*. Number 04; HN29, K5. 1965.

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 2021.

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*, 2009.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

Jiabo Li, Tedman Ehlers, Jon Sutter, Shikha Varma-O'Brien, and Johannes Kirchmair. Caesar: a new conformer generation algorithm based on recursive buildup and local rotational symmetry consideration. *Journal of chemical information and modeling*, 2007.

Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 2021.

Oscar Méndez-Lucio, Mazen Ahmad, Ehecatl Antonio del Rio-Chanona, and Jörg Kurt Wegner. A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, 2021.

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, and José Miguel Hernández-Lobato. Bootstrap your flow. *arXiv preprint*, 2021.

Maria A Miteva, Frederic Guyon, and Pierre Tuffery. Frog2: Efficient 3d conformation ensemble generator for small compounds. *Nucleic acids research*, 2010.

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 2001.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

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

Philipp Pracht, Fabian Bohle, and Stefan Grimme. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, 2020.

Martin Quack. How important is parity violation for molecular and biomolecular chirality? *Angewandte Chemie International Edition*, 2002.

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

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, 2021.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 2017.

Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, 2021.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

Hannes Stärk, Octavian-Eugen Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint*, 2018.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

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

Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gomez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, 2021b.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix F
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Conformer generation is useful for many scientific applications, some of which could have negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix B.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code can be found at `https://github.com/gcorso/torsional-diffusion`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix H.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Approximately 2000 GPU-hours on an internal cluster.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] We used the code and data released by GeoMol and GeoDiff released under MIT license and the GEOM datasets released under CC0 1.0 license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See the instructions in our repository `https://github.com/gcorso/torsional-diffusion`.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All data used is open-source.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No personal data is used.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Definitions

Consider a molecular graph $G = (\mathcal{V}, \mathcal{E})$ and its space of possible conformers $\mathcal{C}_G$. A conformer is an assignment $\mathcal{V} \mapsto \mathbb{R}^3$ of each atom to a point in 3D-space, defined up to global rototranslation. For notational convenience, we suppose there is an ordering of nodes such that we can regard a mapping as a vector in $\mathbb{R}^{3n}$ where $n = |\mathcal{V}|$. Then a conformer $C \in \mathcal{C}_G$ is a set of $SE(3)$-equivalent vectors in $\mathbb{R}^{3n}$—that is, $\mathcal{C}_G \cong \mathbb{R}^{3n}/SE(3)$. This defines the space of conformers in terms of *extrinsic* (or Cartesian) coordinates.

An *intrinsic* (or internal) coordinate is a function over $\mathcal{C}_G$—i.e., it is an $SE(3)$-invariant function over $\mathbb{R}^{3n}$. There are four types of such coordinates typically considered:

Bond lengths. For $(a, b) \in \mathcal{E}$, the bond length $l_{ab} \in [0, \infty)$ is defined as $|x_a - x_b|$.

Bond angles. For $a, b, c \in \mathcal{V}$ such that $a, c \in \mathcal{N}(b)$, the bond angle $\alpha_{abc} \in [0, \pi]$ is defined by

$$\cos \alpha_{abc} := \frac{(x_c - x_b) \cdot (x_a - x_b)}{|x_c - x_b||x_a - x_b|} \tag{11}$$

Chirality. For $a \in \mathcal{V}$ with 4 neighbors $b, c, d, e \in \mathcal{N}(a)$, the chirality $z_{abcd} \in \{-1, 1\}$ is defined as

$$z_{abcde} := \operatorname{sign} \det \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_b - x_a & x_c - x_a & x_d - x_a & x_e - x_a \end{pmatrix} \tag{12}$$

Similar quantities are defined for atoms with other numbers of neighbors. Chirality is often considered part of the specification of the molecule, rather than the conformer. See Appendix F.3 for additional discussion on this point.

Torsion angles. For $(b, c) \in \mathcal{E}$, with a choice of reference neighbors $a \in \mathcal{N}(b) \setminus \{c\}, d \in \mathcal{N}(c) \setminus \{b\}$, the torsion angle $\tau_{abcd} \in [0, 2\pi)$ is defined as the dihedral angle between planes abc and bcd:

$$\begin{aligned} \cos \tau_{abcd} &= \frac{\mathbf{n}_{abc} \cdot \mathbf{n}_{bcd}}{|\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} \\ \sin \tau_{abcd} &= \frac{\mathbf{u}_{bc} \cdot (\mathbf{n}_{abc} \times \mathbf{n}_{bcd})}{|\mathbf{u}_{bc}||\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} \end{aligned} \tag{13}$$

where $\mathbf{u}_{ab} = x_b - x_a$ and $\mathbf{n}_{abc}$ is the normal vector $\mathbf{u}_{ab} \times \mathbf{u}_{bc}$. Note that $\tau_{abcd} = \tau_{dcba}$—i.e., the dihedral angle is the same for four consecutively bonded atoms regardless of the direction in which they are considered.

A **complete set of intrinsic coordinates** of the molecule is a set of such functions $(f_1, f_2, \ldots)$ such that $F(C) = (f_1(C), f_2(C), \ldots)$ is a bijection. In other words, they fully specify a unique element of $\mathcal{C}_G$ without overparameterizing the space. In general there exist many possible such sets for a given molecular graph. We will not discuss further how to find such sets, as our work focuses on manipulating molecules in a way that holds fixed all l, α, z and only modifies (a subset of) torsion angles τ.

As presently stated, the **torsion angle about a bond** $(b, c) \in \mathcal{E}$ is ill-defined, as it could be any τ_{abcd} with $a \in \mathcal{N}(b) \setminus \{c\}, d \in \mathcal{N}(c) \setminus \{b\}$. However, any complete set of intrinsic coordinates needs to only have at most one such τ_{abcd} for each bond (b, c) [Ganea et al., 2021]. Thus, we often refer to *the* torsion angle about a bond (b_i, c_i) as τ_i when reference neighbors a_i, b_i are not explicitly stated.

B Propositions

B.1 Torsion update

Given a freely rotatable bond (b_i, c_i), by definition removing (b_i, c_i) creates two connected components $\mathcal{V}(b_i), \mathcal{V}(c_i)$. Then, consider torsion angle τ_j at a different bond (b_j, c_j) with neighbor choices $a_j \in \mathcal{N}(b_j), d_j \in \mathcal{N}(c_j), a_j \neq c_j, d_j \neq b_j$. Without loss of generality, there are two cases

- Case 1: $a_j, b_j, c_j, d_j \in \mathcal{V}(b_i)$
- Case 2: $d_j \in \mathcal{V}(c_i)$ and $a_j, b_j, c_j \in \mathcal{V}(b_i)$

Note that in Case 2, $c_j = b_i$ and $d_j = c_i$ must hold because there is only one edge between $\mathcal{V}(b_i), \mathcal{V}(c_i)$. With these preliminaries we now restate the proposition:

Proposition 1. *Let (b_i, c_i) be a rotatable bond, let $\mathbf{x}_{\mathcal{V}(b_i)}$ be the positions of atoms on the b_i side of the molecule, and let $R(\boldsymbol{\theta}, x_{c_i}) \in SE(3)$ be the rotation by Euler vector $\boldsymbol{\theta}$ about x_{c_i}. Then for $C, C' \in \mathcal{C}_G$, if τ_i is any definition of the torsion angle around bond (b_i, c_i),*

$$\begin{aligned} \tau_i(C') &= \tau_i(C) + \theta \\ \tau_j(C') &= \tau_j(C) \quad \forall j \neq i \end{aligned} \qquad \textit{if} \qquad \exists \mathbf{x} \in C, \mathbf{x}' \in C'. \quad \begin{aligned} \mathbf{x}'_{\mathcal{V}(b_i)} &= \mathbf{x}_{\mathcal{V}(b_i)} \\ \mathbf{x}'_{\mathcal{V}(c_i)} &= R\left(\theta\, \hat{\mathbf{r}}_{b_i c_i}, x_{c_i}\right) \mathbf{x}_{\mathcal{V}(c_i)} \end{aligned} \tag{14}$$

where $\hat{\mathbf{r}}_{b_i c_i} = (x_{c_i} - x_{b_i})/||x_{c_i} - x_{b_i}||$.

Proof. First we show $\tau_i(C') = \tau_i(C) + \theta$, for which it suffices to show $\tau_i(\mathbf{x}') = \tau_i(\mathbf{x}) + \theta$. Because $a_i, b_i \in \mathcal{V}(b_i)$, $x'_{a_i} = x_{a_i}$ and $x'_{b_i} = x_{b_i}$. Since the rotation of $\mathbf{x}_{\mathcal{V}(c_i)}$ is centered at x_{c_i}, we have $x'_{c_i} = x_{c_i}$ as well. Now we consider d_i and $\mathbf{u}'_{cd} = x'_{d_i} - x_{c_i}$. By the Rodrigues rotation formula,

$$\mathbf{u}'_{cd} = \mathbf{u}_{cd} \cos\theta + \frac{\mathbf{n}_{bcd}}{|\mathbf{u}_{bc}|} \sin\theta + \frac{\mathbf{u}_{bc}}{|\mathbf{u}_{bc}|} \left(\frac{\mathbf{u}_{bc}}{|\mathbf{u}_{bc}|} \cdot \mathbf{u}_{cd} \right) (1 - \cos\theta) \tag{15}$$

Then we have

$$\mathbf{n}'_{bcd} = \mathbf{u}_{bc} \times \mathbf{u}'_{cd} = \mathbf{n}_{bcd} \cos\theta - \left(\mathbf{n}_{bcd} \times \frac{\mathbf{u}_{bc}}{|\mathbf{u}_{bc}|} \right) \sin\theta \tag{16}$$

To obtain $|\mathbf{n}'_{bcd}|$, note that since $\mathbf{n}_{bcd} \perp \mathbf{u}_{bc}$,

$$\left| \mathbf{n}_{bcd} \times \frac{\mathbf{u}_{bc}}{|\mathbf{u}_{bc}|} \right| = |\mathbf{n}_{bcd}| \tag{17}$$

which gives $|\mathbf{n}'_{bcd}| = |\mathbf{n}_{bcd}|$. Thus,

$$\begin{aligned} \cos\tau'_i = \frac{\mathbf{n}_{abc} \cdot \mathbf{n}'_{bcd}}{|\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} &= \frac{\mathbf{n}_{abc} \cdot \mathbf{n}_{bcd}}{|\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} \cos\theta - \frac{\mathbf{n}_{abc} \cdot (\mathbf{n}_{bcd} \times \mathbf{u}_{bc})}{|\mathbf{n}_{abc}||\mathbf{n}_{bcd}||\mathbf{u}_{bc}|} \sin\theta \\ &= \cos\tau_i \cos\theta - \sin\tau_i \sin\theta = \cos(\tau_i + \theta) \end{aligned} \tag{18}$$

Similarly,

$$\begin{aligned} \sin\tau'_i = \frac{\mathbf{u}_{bc} \cdot (\mathbf{n}_{abc} \times \mathbf{n}'_{bcd})}{|\mathbf{u}_{bc}||\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} &= \frac{\mathbf{u}_{bc} \cdot (\mathbf{n}_{abc} \times \mathbf{n}_{bcd})}{|\mathbf{u}_{bc}||\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} \cos\theta - \frac{\mathbf{u}_{bc} \cdot (\mathbf{n}_{abc} \times (\mathbf{n}_{bcd} \times \mathbf{u}_{bc}))}{|\mathbf{u}_{bc}|^2|\mathbf{n}_{abc}||\mathbf{n}_{bcd}|} \sin\theta \\ &= \sin\tau_i \cos\theta + \cos\tau_i \sin\theta = \sin(\tau_i + \theta) \end{aligned} \tag{19}$$

Therefore, $\tau'_i = \tau_i + \theta$

Now we show $\tau'_j = \tau_j$ for all $j \neq i$. Consider any such j. For Case 1, $x'_{a_j} = x_{a_j}, x'_{b_j} = x_{b_j}, x'_{c_j} = x_{c_j}, x'_{d_j} = x_{d_j}$ so clearly $\tau'_j = \tau_j$. For Case 2, $x'_{a_j} = x_{a_j}, x'_{b_j} = x_{b_j}, x'_{c_j} = x_{c_j}$ immediately. But because $d_j = c_i$, we also have $x'_{d_j} = x_{d_j}$. Thus, $\tau'_j = \tau_j$. □

B.2 Parity equivariance

Proposition 2. *If $p(\boldsymbol{\tau}(C) \mid L(C)) = p(\boldsymbol{\tau}(-C) \mid L(-C))$, then for all diffusion times t,*

$$\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}(C) \mid L(C)) = -\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}(-C) \mid L(-C)) \tag{20}$$

Proof. From Equation 13 we see that for any torsion τ_i, we have $\tau_i(-C) = -\tau_i(C)$; therefore $\boldsymbol{\tau}_i(-C) = -\boldsymbol{\tau}_i(C)$, which we denote $\boldsymbol{\tau}_-$. Also denote $\boldsymbol{\tau} := \boldsymbol{\tau}(C), p_t(\boldsymbol{\tau}) := p_t(\boldsymbol{\tau} \mid L(C))$ and $p'_t(\boldsymbol{\tau}_-) := p_t(\boldsymbol{\tau}_- \mid L(-C))$. We claim $p_t(\boldsymbol{\tau}) = p'_t(\boldsymbol{\tau}_-)$ for all t. Since the perturbation kernel (equation 3) is parity invariant,

$$\begin{aligned} p'_t(\boldsymbol{\tau}_-) &= \int_{\mathbb{T}^m} p'_0(\boldsymbol{\tau}'_-) p_{t|0}(\boldsymbol{\tau}_- \mid \boldsymbol{\tau}'_-)\, d\boldsymbol{\tau}'_- \\ &= \int_{\mathbb{T}^m} p_0(\boldsymbol{\tau}') p_{t|0}(\boldsymbol{\tau} \mid \boldsymbol{\tau}')\, d\boldsymbol{\tau}'_- = p_t(\boldsymbol{\tau}) \end{aligned} \tag{21}$$

Next, we have

$$\begin{aligned} \nabla_{\boldsymbol{\tau}} \log p'_t(\boldsymbol{\tau}_-) &= \frac{\partial \boldsymbol{\tau}_-}{\partial \boldsymbol{\tau}} \nabla_{\boldsymbol{\tau}^-} \log p'_t(\boldsymbol{\tau}_-) \\ &= -\nabla_{\boldsymbol{\tau}} \log p_t(\boldsymbol{\tau}) \end{aligned} \tag{22}$$

which concludes the proof. □

B.3 Likelihood conversion

Proposition 3. *Let $\mathbf{x} \in C(\boldsymbol{\tau}, L)$ be a centered conformer in Euclidean space. Then,*

$$p_G(\mathbf{x} \mid L) = \frac{p_G(\boldsymbol{\tau} \mid L)}{8\pi^2\sqrt{\det g}} \quad \text{where} \;\; g_{\alpha\beta} = \sum_{k=1}^{n} J_\alpha^{(k)} \cdot J_\beta^{(k)} \tag{23}$$

where the indices α, β are integers between 1 and $m+3$. For $1 \leq \alpha \leq m$, $J_\alpha^{(k)}$ is defined as

$$J_i^{(k)} = \tilde{J}_i^{(k)} - \frac{1}{n}\sum_{\ell=1}^{n} \tilde{J}_i^{(\ell)} \quad \text{with} \;\; \tilde{J}_i^{(\ell)} = \begin{cases} 0 & \ell \in \mathcal{V}(b_i), \\ \frac{\mathbf{x}_{b_i} - \mathbf{x}_{c_i}}{||\mathbf{x}_{b_i} - \mathbf{x}_{c_i}||} \times (\mathbf{x}_\ell - \mathbf{x}_{c_i}), & \ell \in \mathcal{V}(c_i), \end{cases} \tag{24}$$

and for $\alpha \in \{m+1, m+2, m+3\}$ as

$$J_{m+1}^{(k)} = \mathbf{x}_k \times \hat{x}, \qquad J_{m+2}^{(k)} = \mathbf{x}_k \times \hat{y}, \qquad J_{m+3}^{(k)} = \mathbf{x}_k \times \hat{z}, \tag{25}$$

where (b_i, c_i) is the freely rotatable bond for torsion angle i, $\mathcal{V}(b_i)$ is the set of all nodes on the same side of the bond as b_i, and $\hat{x}, \hat{y}, \hat{z}$ are the unit vectors in the respective directions.

Proof. Let M be $(m+3)$-dimensional manifold embedded in $3n$-dimensional Euclidean space formed by the set of all centered conformers with fixed local structures but arbitrary torsion angles and orientation. A natural set of coordinates for M is $q^\alpha = \{\tau_1, \tau_2, \ldots, \tau_m, \omega_x, \omega_y, \omega_z\}$, where τ_i is the torsion angle at bond i and $\omega_x, \omega_y, \omega_z$ define the global rotation about the center of mass:

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k - \frac{1}{n}\sum_{\ell=1}^{n} \tilde{\mathbf{x}}_\ell \quad \text{where} \quad \tilde{\mathbf{x}}_\ell = e^{\Lambda(\omega)}\mathbf{x}'_k, \quad \Lambda(\omega) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}. \tag{26}$$

Here $\mathbf{x}'_k$ is the position of atom k as determined by the torsion angles, without centering or global rotations, and $\omega_x, \omega_y, \omega_z$ are rotation about the x, y, and z axis respectively.

Consider the set of covariant basis vectors

$$\mathbf{J}_\alpha = \frac{\partial \mathbf{x}}{\partial q^\alpha}. \tag{27}$$

and corresponding the covariant components of the metric tensor,

$$g_{\alpha\beta} = \mathbf{J}_\alpha \cdot \mathbf{J}_\beta = \frac{\partial \mathbf{x}}{\partial q^\alpha} \cdot \frac{\partial \mathbf{x}}{\partial q^\beta}. \tag{28}$$

The conversion factor between torsional likelihood and Euclidean likelihood is given by

$$\int \sqrt{\det \mathbf{g}}\, d^3\omega, \tag{29}$$

where $\sqrt{\det \mathbf{g}}\, d^{m+3}q$ is the invariant volume element on M [Carroll, 2019], and the integration over ω marginalizes over the uniform distribution over global rotations. The calculation of Eq. 29 proceeds as follows.

Let the position of the k'th atom be x_k, and let the three corresponding components of $\mathbf{J}_\alpha$ be $J_\alpha^{(k)}$. For $1 \leq i \leq m$, $J_i^{(k)}$is given by

$$J_i^{(k)} = \frac{\partial}{\partial \tau_i}\left(\tilde{\mathbf{x}}_k - \frac{1}{n}\sum_{\ell=1}^{n} \tilde{\mathbf{x}}_\ell\right) = \tilde{J}_i^{(k)} - \frac{1}{n}\sum_{\ell=1}^{n} \tilde{J}_i^{(\ell)} \tag{30}$$

where $\tilde{J}_i^{(k)} := \partial\tilde{\mathbf{x}}_k/\partial\tau_i$ is the displacement of atom k upon an infinitesmal change in the torsion angle τ_i, without considering the change in the center of mass. Clearly $\tilde{J}_i^{(b_i)} = \tilde{J}_i^{(c_i)} = 0$ because neither b_i nor c_i itself is displaced; furthermore, all atoms on the b side of torsioning bond are not displaced, so $J_i^{(k)} = 0$ for all $k \in \mathcal{N}(b_i)$. The remaining atoms, in $\mathcal{N}(c_i)$, are rotated about the axis of the (b_i, c_i) bond. The displacement per infinitesimal $\partial\tau_i$ is given by the cross product of the unit normal along the rotation axis, $(\tilde{x}_{c_i} - \tilde{x}_{b_i})/||\tilde{x}_{c_i} - \tilde{x}_{b_i}||$, with the displacement from rotation axis, $\tilde{x}_k - \tilde{x}_{b_i}$. This cross product yields $J_\alpha^{(k)}$ in Eq. 24, where the tildes are dropped as relative positions do not depend on center of mass. For $\alpha \in \{m+1, m+2, m+3\}$, a similar consideration of the cross product with the rotation axis yields Eq. 25. Finally, since none of the components of the metric tensor depend explicitly on ω, the integration over ω in Eq. 29 is trivial and yields the volume over $SO(3)$ of $8\pi^2$ [Chirikjian, 2011], proving the proposition. □

C Training and inference procedures

Algorithms 2 and 3 summarize, respectively, the training and inference procedures used for torsional diffusion. In practice, during training, we limit K_G to 30 i.e. we only consider the first 30 conformers found by CREST (typically those with the largest Boltzmann weight). Moreover, molecules are batched and an Adam optimizer with a learning rate scheduler is used for optimization. For inference, to fairly compare with other methods from the literature, we follow Ganea et al. [2021] and set K to be twice the number of conformers returned by CREST.

Algorithm 2: Training procedure

Input: molecules $[G_0, ..., G_N]$ each with true conformers $[C_{G,1}, ...C_{G,K_G}]$, learning rate α
Output: trained score model $\mathbf{s}_\theta$
conformer matching process for each G to get $[\hat{C}_{G,1}, ...\hat{C}_{G,K_G}]$;
for *epoch* $\leftarrow 1$ **to** $epoch_{\max}$ **do**
  for G ***in*** $[G_0, ..., G_N]$ **do**
    sample $t \in [0, 1]$ and $\hat{C} \in [\hat{C}_{G,1}, ...\hat{C}_{G,K_G}]$;
    sample $\Delta\boldsymbol{\tau}$ from wrapped normal $p_{t|0}(\cdot \mid \mathbf{0})$ with $\sigma = \sigma_{\min}^{1-t}\,\sigma_{\max}^{t}$;
    apply $\Delta\boldsymbol{\tau}$ to $\hat{C}$;
    predict $\delta\boldsymbol{\tau} = \mathbf{s}_{\theta,G}(\hat{C}, t)$;
    update $\theta \leftarrow \theta - \alpha\nabla_\theta\|\delta\boldsymbol{\tau} - \nabla_{\Delta\boldsymbol{\tau}} p_{t|0}(\Delta\boldsymbol{\tau} \mid \mathbf{0})\|^2$;

Algorithm 3: Inference procedure

Input: molecular graph G, number conformers K, number steps N
Output: predicted conformers $[C_1, ...C_K]$
generate local structures by obtaining conformers $[C_1, ...C_K]$ from RDKit;
for C ***in*** $[C_1, ...C_K]$ **do**
  sample $\Delta\boldsymbol{\tau} \sim U[0, 2\pi]^m$ and apply to C to randomize torsion angles;
  for $n \leftarrow N$ **to** 1 **do**
    let $t = n/N,\ g(t) = \sigma_{\min}^{1-t}\,\sigma_{\max}^{t}\sqrt{2\ln(\sigma_{\max}/\sigma_{\min})}$;
    predict $\delta\boldsymbol{\tau} = \mathbf{s}_{\theta,G}(\hat{C}, t)$;
    draw $\mathbf{z}$ from wrapped normal with $\sigma^2 = 1/N$;
    set $\Delta\boldsymbol{\tau} = (g^2(t)/N)\ \delta\boldsymbol{\tau} + g(t)\ \mathbf{z}$;
    apply $\Delta\boldsymbol{\tau}$ to C;

D Score network architecture

Overview To perform the torsion score prediction under these symmetry constraints we design an architecture formed by three components: an embedding layer, a series of K interaction layers and a pseudotorque layer. The pseudotorque layer produces pseudoscalar torsion scores $\delta\tau := \partial \log p/\partial\tau$ for every rotatable bond. Following the notation from Thomas et al. [2018] and Batzner et al. [2022], we represent the node representations as $V_{acm}^{(k,l,p)}$ a dictionary with keys the layer k, rotation order l and parity p that contains tensors with shapes $[|\mathcal{V}|, n_l, 2l+1]$ corresponding to the indices of the node, channel and representation respectively. We use the `e3nn` library [Geiger et al., 2022] to implement our architecture.

Embedding layer In the embedding layer, we build a radius graph $(\mathcal{V}, \mathcal{E}_{r_{\max}})$ around each atom on top of the original molecular graph and generate initial scalar embeddings for nodes $V_a^{(0,0,1)}$ and edges e_{ab} combining chemical properties, sinusoidal embeddings of time $\phi(t)$ [Vaswani et al., 2017]

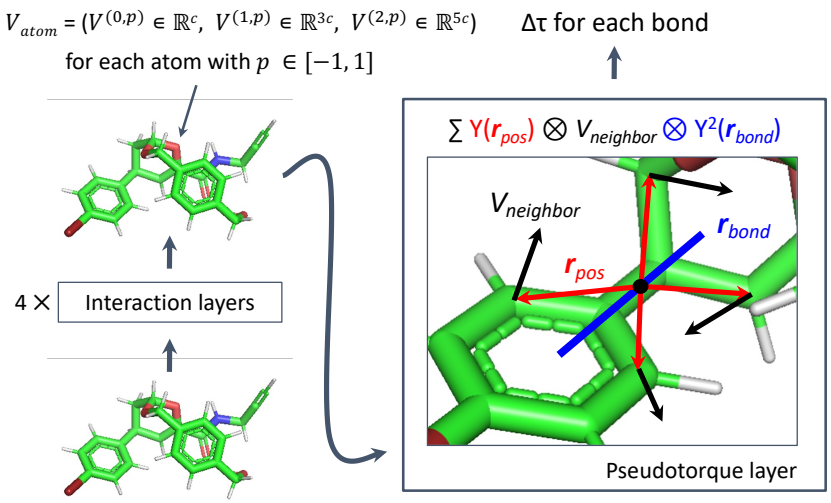

Figure 4: Overview of the architecture and visual intuition of the pseudotorque layer.

and, for the edges, a radial basis function representation of their length $\mu(r_{ab})$ [Schütt et al., 2017]:

$$\begin{aligned} \mathcal{E}_{r_{\max}} &= \mathcal{E} \sqcup \{(a,b) \mid r_{ab} < r_{\max}\} \\ e_{ab} &= \Upsilon^{(e)}(fe_{ab} || \mu(r_{ab}) || \phi(t)) \quad \forall (a,b) \in \mathcal{E}_{r_{\max}} \\ V_a^{(0,0,1)} &= \Upsilon^{(v)}(f_a || \phi(t)) \quad \forall a \in \mathcal{V} \end{aligned} \tag{31}$$

where $\Upsilon^{(e)}$ and $\Upsilon^{(v)}$ are learnable two-layers MLPs, r_{ab} is the Euclidean distance between atoms a and b, $r_{\max} = 5$ Å is the distance cutoff, f_a are the chemical features of atom a, f_{ab} are the chemical features of bond (a, b) if it was part of $\mathcal{E}$ and 0 otherwise.

The node and edge chemical features f_a and f_{ab} are constructed as in Ganea et al. [2021]. Briefly, the node features include atom identity, atomic number, aromaticity, degree, hybridization, implicit valence, formal charge, ring membership, and ring size, constituting a 74-dimensional vector for GEOM-DRUGS and 44-dimensional for QM9 (due to fewer atom types). The edge features are a 4 dimensional one-hot encoding of the bond type.

Interaction layers The interaction layers are based on E(3)NN [Geiger et al., 2022] convolutional layers. At each layer, for every pair of nodes in the graph, we construct messages using tensor products of the current irreducible representation of each node with the spherical harmonic representations of the normalized edge vector. These messages are themselves irreducible representations, which are weighted channel-wise by a scalar function of the current scalar representations of the two nodes and the edge and aggregated with Clebsch-Gordan coefficients.

At every layer k, for every node a, rotation order l_o, and output channel c':

$$\begin{aligned} V_{ac'm_o}^{(k,l_o,p_o)} &= \sum_{l_f,l_i,p_i} \sum_{m_f,m_i} C_{(l_i,m_i)(l_f,m_f)}^{(l_o,m_o)} \frac{1}{|\mathcal{N}_a|} \sum_{b \in \mathcal{N}_a} \sum_{c} \psi_{abc}^{(k,l_o,l_f,l_i,p_i)}\ Y_{m_f}^{(l_f)}(\hat{r}_{ab})\ V_{bcm_i}^{(k-1,l_i,p_i)} \\ &\text{with } \psi_{abc}^{(k,l_o,l_f,l_i,p_i)} = \Psi_c^{(k,l_o,l_f,l_i,p_i)}(e_{ab} || V_a^{(k-1,0,1)} || V_b^{(k-1,0,1)}) \end{aligned} \tag{32}$$

where the outer sum is over values of l_f, l_i, p_i such that $|l_i - l_f| \leq l_o \leq l_i + l_f$ and $(-1)^{l_f} p_i = p_o$, C indicates the Clebsch-Gordan coefficients [Thomas et al., 2018], $\mathcal{N}_a = \{b \mid (a,b) \in \mathcal{E}_{\max}\}$ the neighborhood of a and Y the spherical harmonics. The rotational order of the nodes representations l_o and l_i and of the spherical harmonics of the edges (l_f) are restricted to be at most 2. All the learnable weights are contained in Ψ, a dictionary of MLPs that compute per-channel weights based on the edge embeddings and scalar features of the outgoing and incoming node.

Pseudotorque layer The final part of our architecture is a pseudotorque layer that predicts a pseudoscalar score $\delta\tau$ for each rotatable bond from the per-node outputs of the interaction layers.

For every rotatable bond, we construct a tensor-valued filter, centered on the bond, from the tensor product of the spherical harmonics with a $l = 2$ representation of the *bond axis*. Since the parity of the $l = 2$ spherical harmonic is even, this representation does not require a choice of bond direction. The filter is then used to convolve with the representations of every neighbor on a radius graph, and the products which produce pseudoscalars are passed through odd-function (i.e., with $\tanh$ nonlinearity and no bias) dense layers (not shown in equation 33) to produce a single prediction.

For all rotatable bonds $g = (g_0, g_1) \in \mathcal{E}_{\text{rot}}$ and $b \in \mathcal{V}$, let r_{gb} and $\hat{r}_{gb}$ be the magnitude and direction of the vector connecting the center of bond g and b.

$$
\begin{aligned}
\mathcal{E}_\tau &= \{(g, b) \mid g \in \mathcal{E}_r, b \in \mathcal{V}, r_{gb} < r_{\max}\} \qquad e_{gb} = \Upsilon^{(\tau)}(\mu(r_{gb})) \\
T^{(l_o,p_o)}_{gbm_o} &= \sum_{m_g,m_r,l_r:p_o=(-1)^{l_r}} C^{(l_o,m_o)}_{(2,m_g)(l_r,m_r)} Y^{(2)}_{m_f}(\hat{r}_g)\, Y^{(l_r)}_{m_r}(\hat{r}_{gb}) \\
\delta\tau_g &= \sum_{l,p_f,p_i:p_f p_i=-1} \sum_{m_o,m_i} C^{(0,0)}_{(l,m_f)(l,m_i)} \frac{1}{|\mathcal{N}_g|} \sum_{b\in\mathcal{N}_g} \sum_{c} \gamma^{(l,p_i)}_{gcb}\, T^{(l,p_f)}_{gbm_f}\, V^{(K,l,p_i)}_{bcm_i} \\
&\text{with } \gamma^{(l,p_i)}_{gcb} = \Gamma^{(l,p_i)}_c(e_{gb}||V^{(K,0,1)}_b||V^{(K,0,1)}_{g_0} + V^{(K,0,1)}_{g_1})
\end{aligned}
\tag{33}
$$

where $\Upsilon^{(\tau)}$ and Γ are MLPs with learnable parameters and $\mathcal{N}_g = \{b \mid (g, b) \in \mathcal{E}_\tau\}$.

E Conformer matching

The conformer matching procedure, summarised in Algorithm 4, proceeds as follows. For a molecule with K conformers, we first generate K random local structure estimates $\hat{L}$ from RDKit. To match with the ground truth local structures, we compute the cost of matching each true conformer C with each estimate $\hat{L}$ (i.e. a $K \times K$ cost matrix), where the cost is the best RMSD that can be achieved by modifying the torsions of the RDKit conformer with local structure $\hat{L}$ to match the ground truth conformer C. Note that in practice, we compute an upper bound to this optimal RMSD using the fast von Mises torsion matching procedure proposed by Stärk et al. [2022].

We then find an optimal matching of true conformers C to local structure estimates $\hat{L}$ by solving the linear sum assignment problem over the approximate cost matrix [Crouse, 2016]. Finally, for each matched pair, we find the true optimal $\hat{C}$ by running a differential evolution optimization procedure over the torsion angles [Méndez-Lucio et al., 2021]. The complete assignment resulting from the linear sum solution guarantees that there is no distributional shift in the local structures seen during training and inference.

Algorithm 4: Conformer matching

Input: true conformers of G $[C_1, ...C_K]$
Output: approximate conformers for training $[\hat{C}_1, ...\hat{C}_K]$
generate local structures $[\hat{L}_1, ...\hat{L}_K]$ with RDKit;
for (i, j) ***in*** $[1, K] \times [1, K]$ **do**
  C_{temp} = von_Mises_matching(C_i, $\hat{L}_j$);
  cost[i,j] = $\text{RMSD}(C_i, C_{\text{temp}})$;
assignment = linear_sum_assignment(cost);
for $i \leftarrow 1$ **to** K **do**
  j = assignment[i];
  $\hat{C}_i$ = differential_evolution(C_i, $\hat{L}_j$, RMSD);

Table 5 shows the average RMSD between a ground truth conformer C_i and its matched conformer $\hat{C}_i$. The average RMSD of 0.324 Å obtained via conformer matching provides an approximate lower bound on the achievable AMR performance for methods that do not change the local structure and take those from RDKit (further discussion in Appendix F.1).

Table 5: Average $\mathrm{RMSD}(c_i, \hat{c}_i)$ achieved by different variants of conformer matching. "Original RDKit" refers to the RMSD between a random RDKit conformer and a ground truth conformer without any optimization. In "Von Mises optimization" and "Differential evolution," the torsions of the RDKit conformer are adjusted using the respective procedures, but the pairing of RDKit and ground truth conformers is still random. In "Conformer matching," the cost-minimizing assignment prior to differential evolution provides a 15% improvement in average RMSD. The results are shown for a random 300-molecule subset of GEOM-DRUGS.

Matching method	RMSD (Å)
Original RDKit	1.448
Von Mises optimization	0.728
Differential evolution	0.379
Conformer matching	0.324

F Additional discussion

F.1 RDKit local structures

In this section, we provide empirical justification for the claim that cheminformatics methods like RDKit already provide accurate local structures. It is well known in chemistry that bond lengths and angles take on a very narrow range of values due to strong energetic constraints. However, it is not trivial to empirically evaluate the claim due to the difficulty in defining a distance measure between a pair of local structures. In this section, we will employ two sets of observations: marginal error distributions and matched conformer RMSD.

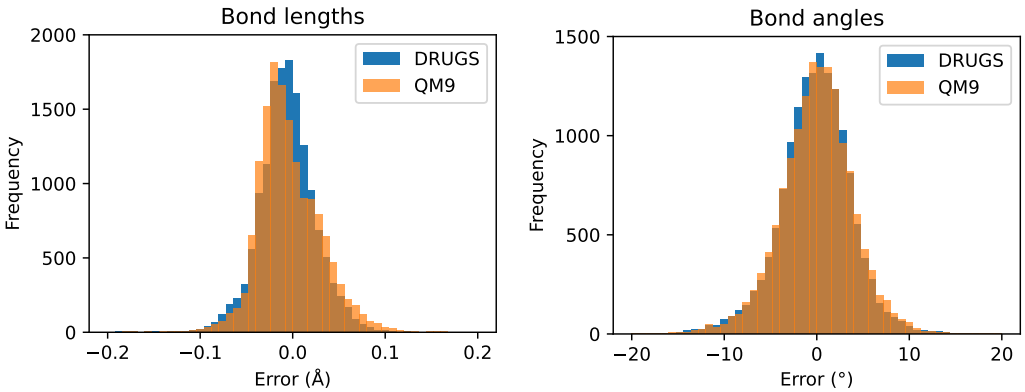

Figure 5: Histogram of the errors in 15000 predicted bond lengths and angles from randomly sampled molecules in GEOM-DRUGS and GEOM-QM9.

Marginal error distributions We examine the distribution of errors of the bond lengths and angles in a random RDKit conformer relative to the corresponding lengths and angles in a random CREST conformer (Figure 5). The distributions are narrow and uni-modal distributions around zero, with a RMSE of 0.03 Å for bond lengths and 4.1° for bond angles on GEOM-DRUGS. Comparing DRUGS and QM9, the error distribution does not depend on the size of the molecule. Although it is difficult to determine how these variations will compound or compensate for each other in the global conformer structure, the analysis demonstrates that bond lengths and angles have little flexibility (i.e., no strong variability among conformers) and are accurately predicted by RDKit.

Matched conformer RMSD We can more rigorously analyze the quality of a local structure $\hat{L}$ with respect to a given reference conformer C by computing the minimum RMSD that can be obtained by combining $\hat{L}$ with optimal torsion angles. That is, we consider the RMSD distance of C to the closest point on the manifold of possible conformers with local structure $\hat{L}$: $\mathrm{RMSD}_{\min}(C, \hat{L}) := \min_\tau \mathrm{RMSD}(C, \hat{C})$ where $\hat{C} = (\hat{L}, \tau)$.

Conveniently, $\hat{C}$ is precisely the output of the differential evolution in Appendix E. Thus, the average RSMD reported in the last row of Table 5 is the expected $\mathrm{RMSD}_{\min}$ of an optimal assignment of RDKit local structures to ground-truth conformers. This distance—0.324 Å on GEOM-DRUGS—is significantly smaller than the error of the current state-of-the-art conformer generation methods. Further, it is only slightly larger than the average $\mathrm{RMSD}_{\min}$ of 0.284 Å resulting from matching a ground truth conformer to the local structure of another randomly chosen ground truth conformer, which provides a measure of the variability among ground truth local structures. These observations support the claim that the accuracy of existing approaches on drug-like molecules can be significantly improved via better conditional sampling of torsion angles.

F.2 Torsion updates

In the main text, we viewed updates $\Delta\tau$ as changes to a torsion angle τ, and asserted that the same update applied to any torsion angle at a given bond (i.e., with any choice of reference neighbors) results in the same conformer. Given this, a potentially more intuitive presentation is to *define* $\Delta\tau$ for a bond as a relative rotation around that bond, without reference to any torsion angle.

Consider a rotatable bond (b, c) and the connected components $\mathcal{V}(b), \mathcal{V}(c)$ formed by removing the bond. Let $\hat{\mathbf{r}}_{bc} = (x_c - x_b)/|x_c - x_b|$ and similarly $\hat{\mathbf{r}}_{cb} = -\hat{\mathbf{r}}_{bc}$. Because the bond is freely rotatable, consider rotations of each side of the molecule around the bond axis given by $\hat{\mathbf{r}}_{bc}$. Specifically, let $\mathcal{V}(b)$ be rotated by some Euler vector $\boldsymbol{\theta}_b := \theta_b \hat{\mathbf{r}}_{bc}$ around x_b, and $\mathcal{V}(c)$ by $\boldsymbol{\theta}_c := \theta_c \hat{\mathbf{r}}_{bc}$ around x_c. Then the rotations induce a *torsion update* $\Delta\tau$ if $\theta_c - \theta_b = \Delta\tau$; or equivalently

$$\Delta\tau = (\boldsymbol{\theta}_c - \boldsymbol{\theta}_b) \cdot \hat{\mathbf{r}}_{bc} \tag{34}$$

The expression remains unchanged if we swap the indices b, c; thus there is no sign ambiguity. Some less formal but possibly more intuitive restatements of the sign convention are:

- Looking down a bond, a *positive* update is given by a CCW rotation of the nearer side; or a CW rotation of the further side
- For a viewer positioned in the middle of the bond, a *positive* update is given by the CW rotation of any one side
- A *positive* update is given by Euler vectors that point *outwards* from the bond

These are illustrated in Figure 6.

Since the Euler vector $\boldsymbol{\theta}$ is a pseudovector that remains unchanged under parity inversion, while $\hat{\mathbf{r}}$ is a normal vector, it is apparent that $\Delta\tau$—and any model predicting $\Delta\tau$—must be a pseudoscalar.

Because the update is determined by a relative rotation, it is not necessary to specify which side to rotate. That is, the same torsion update can be accomplished by rotating only one side, both sides in opposite directions, or both sides in the same direction. In practical implementation, we rotate the side of the molecule with fewer atoms, and keep the other side fixed.

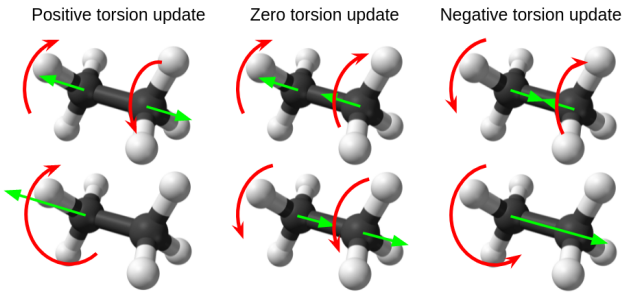

Figure 6: Torsion updates as relative rotations, with the rotations shown with curved red arrows and Euler vectors shown with straight green arrows. The second row emphasizes that the sign convention and update depend only on the *relative* motion of the two sides.

F.3 Chemical isomerism

We have defined a molecule in terms of its bond connectivity, i.e., as a graph $G = (\mathcal{V}, \mathcal{E})$ with atoms $\mathcal{V}$ and $\mathcal{E}$. In chemistry, however, it is universal to consider molecules with the same connectivity, but whose conformers cannot interconvert, as different molecules called *stereoisomers*. In our formalism, stereoisomers correspond to subsets of the space of conformers $\mathcal{C}_G$ for some molecular graph G. Many types of stereoisomerism exist, but the two most important are:

- **Chirality**. Conformers with distinct values of chirality tags $\{z_i\}$—one for each chiral atom meeting certain criteria—are considered different molecules.
- **E/Z isomerism**, also called cis/trans isomerism. For each double bond meeting certain criteria, the space $[0, 2\pi)$ is partitioned into two halves, such that conformers are considered different molecules depending on the value of the torsion angle.

These are not meant to be formal definitions, and we refer to standard chemistry texts for a more detailed treatment. For our purposes, the key implication is that conformer generation requires generating conformers consistent with a *given stereoisomer*. For a molecular graph G with k relevant chiral centers and l relevant double bonds, there are 2^{k+l} possible stereoisomers, corresponding to the partition of $\mathcal{C}_G$ into 2^{k+l} disjoint subsets—only one of which corresponds to the molecule under consideration.

Torsional diffusion automatically handles chirality. Because we have considered chirality to be part of the local structure, it is drawn from the cheminformatics package RDKit, which is given the full identification of the stereoisomer along with the molecular graph, and is not modified by the torsional diffusion. Hence, our method always generates conformers with the correct chirality at each chiral center. On the other hand, GeoDiff does not consider chirality at all, while GeoMol generates molecules without any chirality constraints, and merely inverts the chiral centers that were generated incorrectly.

E/Z isomerism is significantly trickier, as it places a constraint on the torsion angles at double bonds, which are considered freely rotatable in our framework. Presently, torsional diffusion does not attempt to capture E/Z isomerism. One possible way of doing so is to augment the molecular graph with edges of a special type, and we leave such augmentation to future work. GeoDiff and GeoMol also do not attempt to treat E/Z isomerism.

More generally, while the abstract view of molecules as graphs has enabled rapid advances in molecular machine learning, stereoisomerism shows that it is clearly a simplification. As stereoisomers can have significantly different chemical properties and bioactivities, a more complete view of molecular space will be essential for further advances in molecular machine learning.

F.4 Limitations of torsional diffusion

As demonstrated in Section 4, torsional diffusion significantly improves the accuracy and reduces the denoising runtime for conformer generation. However, torsional diffusion also has a number of limitations that we will discuss in this section.

Conformer generation The first clear limitation is that the error that torsional diffusion can achieve is lower bounded by the *quality of the local structure* from the selected cheminformatics method. As discussed in Appendix F.1, this corresponds to the mean RMSD obtained after conformer matching, which is 0.324 Å with RDKit local structures on DRUGS. Moreover, due to the the local structure *distributional shift* discussed in Section 4.1, conformer matching (or another method bridging the shift) is required to generate the training set. However, the resulting conformers are not the minima of the (unconditional or even conditional) potential energy function. Thus, the learning task becomes less physically interpretable and potentially more difficult; empirically we observe this clearly in the training and validation score-matching losses. We leave to future work the exploration of *relaxations* of the rigid local structures assumption in a way that would still leverage the predominance of torsional flexibility in molecular structures, while at the same time allowing some flexibility in the independent components.

Rings The largest source of flexibility in molecular conformations that is not directly accounted for by torsional diffusion is the variability in *ring conformations*. Since the torsion angles at bonds inside cycles cannot be independently varied, our framework treats them as part of the local structure. Therefore, torsional diffusion relies on the local structure sampler $p_G(L)$ to accurately model cycle conformations. Although this is true for a large number of relatively small rings (especially aromatic ones) present in many drug-like molecules, it is less true for puckered rings, fused rings, and larger cycles. In particular, torsional diffusion does not address the longstanding difficulty that existing cheminformatics methods have with macrocycles—rings with 12 or more atoms that have found several applications in drug discovery [Driggers et al., 2008]. We hope, however, that the idea of restricting diffusion processes to the main sources of flexibility will motivate future work to define diffusion processes over cycles conformations combined with free torsion angles.

Boltzmann generation With Boltzmann generators we are typically interested in sampling the Boltzmann distribution over the entire (Euclidean) conformational space $p_G(C)$. However, the procedure detailed in Section 3.6 generates (importance-weighted) samples from the Boltzmann distribution *conditioned* on a given local structure $p_G(C \mid L)$. To importance sample from the full Boltzmann distribution $p_G(C)$, one would need a model $p_G(L)$ over local structures that also provides exact likelihoods. This is not the case with RDKit or, to the best of our knowledge, other existing models, and therefore an interesting avenue for future work.

Proteins As protein conformations are often described with backbone dihedral (i.e., torsion angles), it is natural to consider whether torsional diffusion may be useful for modeling protein flexibility. However, we do not believe that the *direct* application of the framework to proteins or other macromolecules is very promising. Small changes in torsional coordinates cause large displacements in distant regions of the molecule, so the influence on a torsional score is not limited to the local neighborhood of the bond. For small molecules—even the ones in GEOM-XL—this is not a problem because of their limited spatial and graph theoretic diameters. In proteins, however, the graph diameter is 3 times the sequence length and can easily reach over 1000; and interactions between distant residues are extremely important in determining the structure and constraining flexibility. Although torsional diffusion may not be the right framework for modeling proteins, we believe that similar ideas (i.e., well-chosen diffusions over the flexible degrees of freedom) could be useful for generative models of protein structure and is a promising avenue of work.

G Experimental details

G.1 Dataset details

Splits We follow the data processing and splits from Ganea et al. [2021]. The splits are random with train/validation/test of 243473/30433/1000 for GEOM-DRUGS and 106586/13323/1000 for GEOM-QM9. GEOM-XL consists of only a test split (since we do not train on it), which consists of all 102 molecules in the MoleculeNet dataset with at least 100 atoms. For all splits, the molecules whose CREST conformers all have a canonical SMILES different from the SMILES of the molecule (meaning a reacted conformer), or that cannot be handled by RDKit, are filtered out.

Dataset statistics As can be seen in Figure 7, the datasets differ significantly in molecule size as measured by number of atoms or rotatable bonds. Particularly significant is the domain shift between DRUGS and XL, which we leverage in our experiments by testing how well models trained on DRUGS generalize to XL.

Boltzmann generator The torsional Boltzmann generator described in Section 4.5 is trained and tested on molecules from GEOM-DRUGS with 3–7 rotatable bonds. The training (validation) set consists of 10000 (400) such randomly selected molecules from the DRUGS training (validation) set. The test set consists of all the 453 molecules present in the DRUGS test set with 3–7 rotatable bonds.

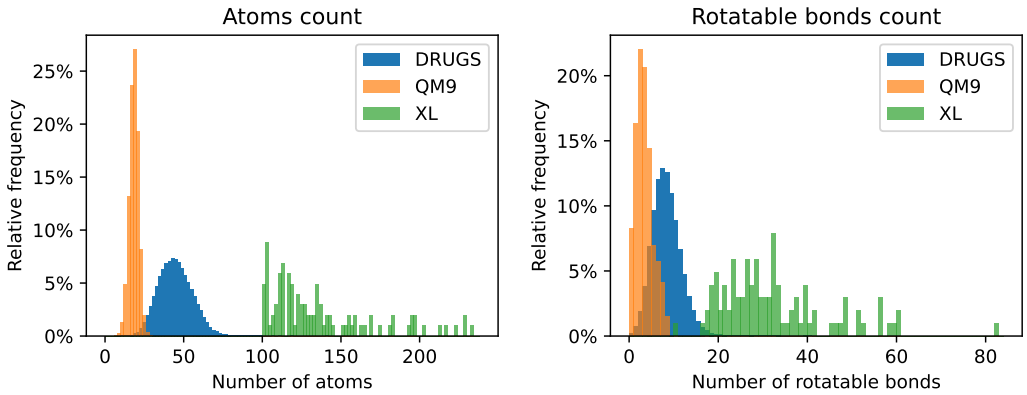

Figure 7: Statistics about the atoms and rotatable bonds counts in the three different datasets.

G.2 Training and tuning details

Conformer generation For conformer ensemble generation on GEOM-DRUGS, the torsional diffusion models were trained on NVIDIA RTX A6000 GPUs for 250 epochs with the Adam optimizer (taking from 4 to 11 days on a single GPU). The hyperparameters tuned on the validation set were (in bold the value that was chosen): initial learning rate (0.0003, **0.001**, 0.003), learning rate scheduler patience (5, **20**), number of layers (2, **4**, 6), maximum representation order (1st, **2nd**), $r_{\max}$ (**5Å**, 7Å, 10Å) and batch norm (**True**, False). All the other default hyperparameters used can be found in the attached code. For GEOM-XL the same trained model was used; for GEOM-QM9 a new model with the same hyperparameters was trained.

Torsional Boltzmann generators We start from a torsional diffusion model pretrained on GEOM-DRUGS, and train for 250 epochs (6-9 days on a single GPU). A separate model is trained for every temperature. The resampling procedure with 5 steps is run for every molecule every $\max(5, ESS)$ epochs, where ESS is computed for the current set of 32 samples. The only hyperparameter tuned (at temperature 300K) is $\sigma_{\min}$, the noise level at which to stop the reverse diffusion process.

We further improve the training procedure of torsional Boltzmann generators by implementing *annealed training*. The Boltzmann generator for some temperature T is trained at epoch k by using the Boltzmann distribution at temperature $T' = T + (3000 - T)/k$ as the target distribution for that epoch. Intuitively, this trains the model at the start with a smoother distribution that is easier to learn, which gradually transforms into the desired distribution.

G.3 Evaluation details

Ensemble RMSD As evaluation metrics for conformer generation, Ganea et al. [2021] and following works have used the so-called Average Minimum RMSD (AMR) and Coverage (COV) for Precision (P) and Recall (R) measured when generating twice as many conformers as provided by CREST. For $K = 2L$ let $\{C_l^*\}_{l\in[1,L]}$ and $\{C_k\}_{k\in[1,K]}$ be respectively the sets of ground truth and generated conformers:

$$\begin{aligned} \text{COV-R} &:= \frac{1}{L}\left|\{l \in [1..L] : \exists k \in [1..K], \text{RMSD}(C_k, C_l^*) < \delta\right| \\ \text{AMR-R} &:= \frac{1}{L}\sum_{l\in[1..L]} \min_{k\in[1..K]} \text{RMSD}(C_k, C_l^*) \end{aligned} \tag{35}$$

where δ is the coverage threshold. The precision metrics are obtained by swapping ground truth and generated conformers.

In the XL dataset, due to the size of the molecules, we compute the RMSDs without testing all possible symmetries of the molecules, therefore the obtained RMSDs are an upper bound, which we find to be very close in practice to the permutation-aware RSMDs.

Runtime evaluation We benchmark the methods on CPU (Intel i9-9920X) to enable comparison with RDKit. The number of threads for RDKit, `numpy`, and `torch` is set to 8. We select 10 molecules at random from the GEOM-DRUGS test set and generate 8 conformers per molecule using each method. Script loading and model loading times are not included in the reported values.

Boltzmann generator To evaluate how well the torsional Boltzmann generator and the AIS baselines sample from the conditional Boltzmann distribution, we report their median effective sample size (ESS) [Kish, 1965] given the importance sampling weights w_i of 32 samples for each molecule:

$$ESS = \frac{\left(\sum_{i=1}^{32} w_i\right)^2}{\sum_{i=1}^{32} w_i^2} \tag{36}$$

This approximates the number of independent samples that would be needed from the target Boltzmann distribution to obtain an estimate with the same variance as the one obtained with the importance-weighted samples.

For the baseline annealed importance samplers, the transition kernel is a single Metropolis-Hastings step with the wrapped normal distributions on $\mathbb{T}^m$ as the proposal. We run with a range of kernel variances: $0.25, 0.5, 0.3, 0.5, 0.75, 1, 1.5.2$; and report the best result. We use an exponential annealing schedule; i.e., $p_n \propto p_0^{1-n/N} p_N^{n/N}$ where p_0 is the uniform distribution and p_N is the target Boltzmann density.

H Additional results

Performance vs size Figure 8 shows the performance of different models as a function of the number of rotatable bonds. Molecules with more rotatable bonds are more flexible and are generally larger; it is therefore expected that the RMSD error will increase with the number of bonds. With very few rotatable bonds, the error of torsional diffusion depends mostly on the quality of the local structures it was given, and therefore it has a similar error as RDKit. However, as the number of torsion angles increases, torsional diffusion deteriorates more slowly than other methods.

The trend continues with the very large molecules in GEOM-XL (average 136 atoms and 32 rotatable bonds). These not only are larger and more flexible, but—for machine learning models trained on GEOM-DRUGS—are also out of distribution. As shown in Table 6, on GEOM-XL GeoMol only performs marginally better than RDKit, while torsional diffusion reduces RDKit AMR by 30% on recall and 12% on precision. These results can very likely be improved by training and tuning the torsional diffusion model on larger molecules.

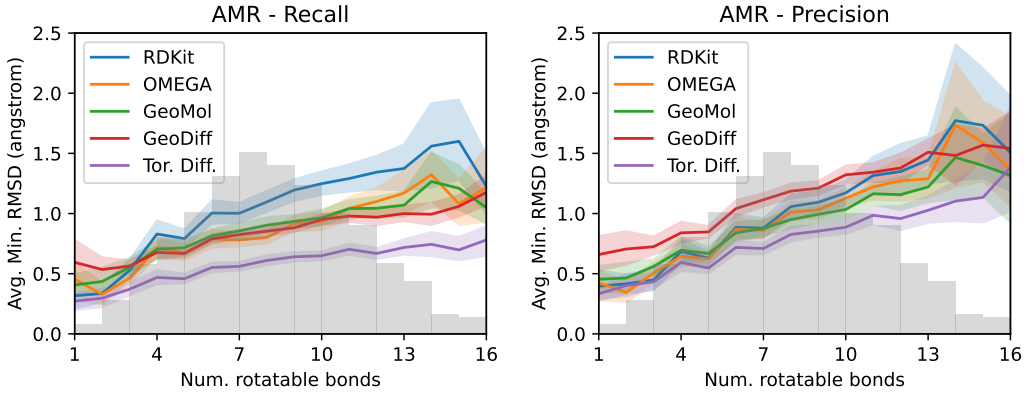

Figure 8: Average minimum RMSD (AMR) for recall (*left*) and precision (*right*) of the different conformer generation methods for molecules with different number of rotatable bonds in GEOM-DRUGS. The background shows the distribution of the number of rotatable bonds.

Table 6: Performance of various methods on the GEOM-XL dataset.

Model	AMR-R ↓ Mean	AMR-R ↓ Med	AMR-P ↓ Mean	AMR-P ↓ Med
RDKit	2.92	2.62	3.35	3.15
GeoMol	2.47	2.39	3.30	3.15
Torsional Diffusion	**2.05**	**1.86**	**2.94**	**2.78**

Table 7: Performance of various methods on the GEOM-QM9 dataset test-set ($\delta = 0.5$Å). Again GeoDiff was retrained on the splits from Ganea et al. [2021].

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
RDKit	85.1	**100.0**	0.235	0.199	86.8	**100.0**	0.232	0.205
OMEGA	85.5	**100.0**	**0.177**	**0.126**	82.9	**100.0**	0.224	**0.186**
GeoMol	91.5	**100.0**	0.225	0.193	86.7	**100.0**	0.270	0.241
GeoDiff	76.5	**100.0**	0.297	0.229	50.0	33.5	0.524	0.510
Torsional diffusion	**92.8**	**100.0**	0.178	0.147	**92.7**	**100.0**	**0.221**	0.195

Small molecules We also train and evaluate our model on the small molecules from GEOM-QM9 and report the performance in Table 7. For these smaller molecules, cheminformatics methods already do very well and, given the very little flexibility and few rotatable bonds present, the accuracy of local structure significantly impacts the performance of torsional diffusion. RDKit achieves a mean recall AMR just over 0.23Å, while torsional diffusion based on RDKit local structures results in a mean recall AMR of 0.178Å. This is already very close lower bound of 0.17Å that can be achieved with RDKit local structures (as approximately calculated by conformer matching). Torsional diffusion does significantly better than other ML methods, but is only on par with or slightly worse than OMEGA, which, evidently, has a better local structures for these small molecules.

Ablation experiments In Table 8 we present a set of ablation studies to evaluate the importance of different components of the proposed torsional diffusion method:

1. *Baseline* refers to the model described and tested throughout the paper.
2. *Probability flow ODE* refers to using the ODE formulation of the reverse diffusion process (not an ablation, strictly speaking). As expected, it obtains similar results to the baseline SDE formulation.
3. *Only D.E. matching* refers to a model trained on conformers obtained by a random assignment of RDKit local structures to ground truth conformers (without first doing an optimal assignment as in Appendix E); this performs only marginally worse than full conformer matching.
4. *First order irreps* refers to the same model but with node irreducible representations kept only until order $\ell = 1$ instead of $\ell = 2$; this worsens the average error by about 5%, but results in a 41% runtime speed-up.
5. *Train on ground truth L* refers to a model trained directly on the ground truth conformers without conformer matching but tested (as always) on RDKit local structures; although the training and validation score matching loss of this model is significantly lower, its inference performance reflects the detrimental effect of the local structure distributional shift.
6. *No parity equivariance* refers to a model whose outputs are parity invariant instead of parity equivariant; the model cannot distinguish a molecule from its mirror image and fails to learn, resulting in performance on par with a random baseline.
7. *Random $\boldsymbol{\tau}$* refers to a random baseline using RDKit local structures and uniformly random torsion angles.

Table 8: Ablation studies with ensemble RMSD on GEOM-DRUGS. Refer to the text in the Appendix for an explanation of each entry. As usual, we compute Coverage with $\delta = 0.75$ Å.

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
Baseline	72.7	80.0	0.582	0.565	55.2	**56.9**	**0.778**	**0.729**
Probability flow ODE	**73.1**	80.4	**0.577**	**0.557**	**55.3**	55.7	0.779	0.737
Only D.E. matching	72.5	**81.1**	0.588	0.569	53.8	56.1	0.794	0.749
First order irreps	70.1	77.9	0.605	0.589	51.4	51.4	0.817	0.783
Train on ground truth L	34.8	22.4	0.920	0.909	22.3	7.8	1.182	1.136
No parity equivariance	30.5	12.5	0.928	0.929	17.9	3.9	1.234	1.217
Random $\boldsymbol{\tau}$	30.9	13.2	0.922	0.923	18.2	4.0	1.228	1.217

Table 9: Ensemble RMSD results on GEOM-DRUGS for varying number of diffusion steps. 20 steps were used for all results reported elsewhere. As usual, we compute Coverage with $\delta = 0.75$ Å.

	Recall				Precision			
	Coverage ↑		AMR ↓		Coverage ↑		AMR ↓	
Steps	Mean	Med	Mean	Med	Mean	Med	Mean	Med
3	42.9	33.8	0.820	0.821	24.1	11.1	1.116	1.100
5	58.9	63.6	0.698	0.685	35.8	26.6	0.979	0.963
10	70.6	78.8	0.600	0.580	50.2	48.3	0.827	0.791
20	72.7	80.0	0.582	0.565	55.2	56.9	0.778	0.729
50	73.1	80.4	0.578	0.557	57.6	60.7	0.753	0.699

Table 10: Median absolute error of generated v.s. ground truth ensemble properties with and without relaxation. $E, \Delta\epsilon, E_{\min}$ in kcal/mol, μ in debye.

	Without relaxation				With relaxation			
Method	E	μ	$\Delta\epsilon$	$E_{\min}$	E	μ	$\Delta\epsilon$	$E_{\min}$
RDKit	39.08	1.40	5.04	39.14	0.81	0.52	0.75	1.16
OMEGA	**16.47**	**0.78**	**3.25**	**16.45**	0.68	0.66	0.68	0.69
GeoMol	43.27	1.22	7.36	43.68	0.42	**0.34**	0.59	0.40
GeoDiff	18.82	1.34	4.96	19.43	0.31	0.35	0.89	0.39
Tor. Diff.	36.91	0.92	4.93	36.94	**0.22**	0.35	**0.54**	**0.13**

Reverse diffusion steps In Table 9 we vary the number of steps used in the reverse diffusion process and evaluate the ensemble RMSD results on GEOM-DRUGS. We find that torsional diffusion is remarkably parsimonious in terms of number of steps required: the majority of gain in performance over prior diffusion-based methods is attained with only 10 steps. We confirm that increasing the number of steps from the default of 20 to 50 only results in minor performance gains.

Ensemble properties In Table 10, we report the median absolute errors of the Boltzmann-weighted properties of the generated vs CREST ensembles, with and without GFN2-xTB relaxation. For all methods, the errors without relaxation are far too large for the computed properties to be chemically useful—for reference, the thermal energy at room temperature is 0.59 kcal/mol. In realistic settings, relaxation of local structures is necessary for any method, after which errors from global flexibility become important. After relaxation, torsional diffusion obtains property approximations on par or better than all previous methods.

Torsional Boltzmann generator Figure 9 shows the histograms of ESSs at 500K for the torsional Boltzmann generator and the AIS baseline. While AIS fails to generate more than one effective sample for most molecules (tall leftmost column), torsional Boltzmann generators are much more efficient, with more than five effective samples for a significant fraction of molecules.

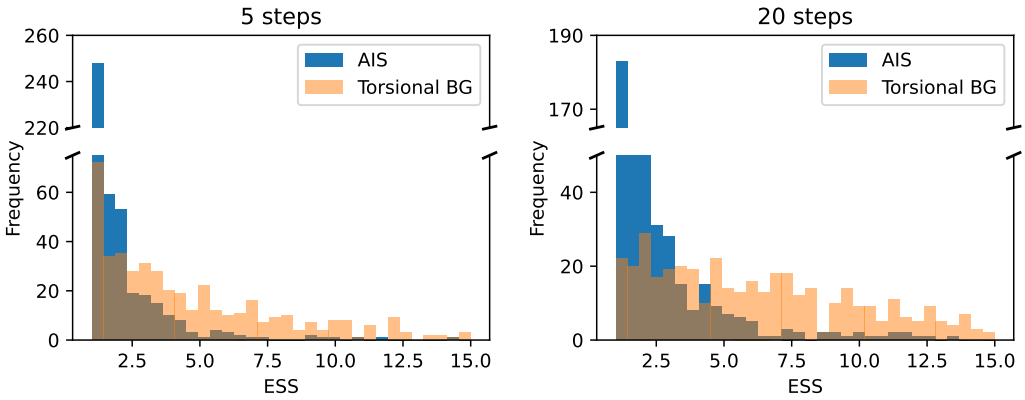

Figure 9: Histogram of the ESSs of the torsional Boltzmann generator and AIS baseline at 500K.