Conformer Search Using SE3-Transformers and Imitation Learning

Luca Thiede *
University of Toronto

Santiago Miret *
Intel Labs

Krzysztof Sadowski
Intel Poland

Haoping Xu
University of Toronto

Mariano Phielipp
Intel Labs

Alan Aspuru-Guzik
University of Toronto

Abstract

We introduce a novel approach to conformer search, the discovery of three-dimensional structures for two-dimensional molecular formulas. We focus on organic molecules using deep imitation learning and equivariant graph neural networks, with the prospect of using reinforcement learning algorithms for fine tuning. To that end, we present our interactive environment that describes the molecule in a rigid-rotor approximation and leverage a behavioral cloning torsion policy to autoregressively determine the dihedral angles of the molecule ultimately yielding a three-dimensional molecular structure. For our policy architecture, we leverage an SE(3) equivariant neural network, which enables us to exploit inherent molecular symmetries and to respect the topology of the angle distribution using a Mixture of Projected Normals action distribution. Our preliminary results for a policy trained on a behavioral cloning objective using the QM9 dataset for expert trajectories shows that the policy can accurately predict torsion angles for various molecules. We believe this to be a promising starting point for future work pertaining to performing conformer search using deep reinforcement learning.

1 Introduction

Many molecular properties heavily depend on the 3D arrangement of the atoms in space. Thus, predicting the structure of a given molecule is a fundamental question of computational chemistry. In 3D molecular structures, the energy of a given state governs its likelihood: The lower the energy, the more likely we are to find the molecule in that given arrangement, where different low energy states of a given molecule are known as its conformers. Generally, the likelihood of a given state with energy E_s at temperature T is given by $p(s) = \frac{e^{-\frac{E_s}{k_b T}}}{Z}$. Yet, modelling $p(s)$ is extremely difficult, as it involves estimating the normalization constant Z. Recent works have shown substantial progress in applying using machine learning by learning $p(s)$ directly from data [Mansimov et al., 2019, Ganea et al., 2021, Jing et al., 2022, Xu et al., 2021]. However, the primary drawback of the aforementioned models' reliance on data is, that it is uncertain how well they generalize to out-of-distribution molecules. Ideally one would like to have a way to incorporate the energy E_s directly, which can be gained from ab-initio physical simulations to train a generalizeable conformer search policy.

In order to simplify the problem of searching the entire chemical space of $p(s)$, we consider only the k lowest energy modes ${s^*}_{0:k}$ for organic molecules in three dimensional space. Many algorithms have been developed for this simplified conformer search problem, including rdkit's Experimental-Torsion

*Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022) AI4Mat Workshop.

Distance Geometry with Basic Knowledge [Landrum] and OpenBabel's Confab systematic search algorithm [O'Boyle et al., 2011]. The former works by defining a set of known constraints on atom distances. It then samples many configurations that fulfil these constraints and picks the ones that minimize energy. OpenBabel, on the other hand, decomposes the molecule into rigid fragments with distances and angles taken from experimental data and assumes that these are fixed, which is called the rigid-rotor approximation. The rigid-rotor approximation leaves only the rotatable bonds as free parameters, over which OpenBabel searches using different strategies. One major shortcoming of these aforedescribed traditional approaches is that they begin every optimization from scratch, even if they have optimized a similar molecule before. In other words, they do not learn from prior experience. Thus, we propose a framework that applies deep imitation and reinforcement learning techniques. Our framework enables us to use energies directly as rewards along with the ability to learn from prior experience using a neural network for more effective training and broader generalization.

2 Method

2.1 Deep Reinforcement Learning & Deep Imitation Learning

Reinforcement Learning (RL) is a technique used to find a policy $\pi_\theta(a_i|s_{i-1})$ parameterized by the parameters θ that interacts with an environment such that the expected reward of state-action trajectories $\tau = (s_0, a_0, s_1, a_1, ..., s_N, a_N)$ sampled from the policy are maximized [Sutton and Barto, 2018]. In other words, we want to find θ^* that maximizes $\mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)]$ where $R(\tau)$ assigns a reward to a trajectory. Given the difficulty to train RL policies from scratch, we pretrain our policy using Imitation Learning (IL). Specifically, we apply behavioural cloning (BC) [Osa et al., 2018] where we create a dataset of state action pairs $s_i, a_{i\,i=1:N}$ obtained from an expert policy, which we approximate with ground truth conformers from the QM9 dataset [Ramakrishnan et al., 2014, Ruddigkeit et al., 2012]. This allows us to frame policy training as supervised learning and minimize the negative log-likelihood of the expert's action, given the state those experts were in. Concretely, we minimize $-\log\sum_i^N \pi_\theta(a_i|s_i)$ over a training dataset of expert trajectories and evaluate the policy on a validation set of trajectories.

2.2 Molecular Design Environment

Interactive learning methods such as RL and IL require an environment to train and deploy the policies. Our custom-built environment tracks the molecular state and updates the atomic coordinates as the policy chooses its actions, adopting an OpenAI Gym [Brockman et al., 2016] interface. In our formulation, we define the state as a 'query graph' that describes the atoms and bonds of the molecule, but not the 3D coordinates. The generation process then starts by randomly picking a starting fragment from the query graph. Next, we randomly select a new fragment from the query graph that is attached to the previously placed one defined as s_0. subsequently, s_0 serves as input to an SE(3)-Transformer [Fuchs et al., 2020] backbone to predict the angle α_0 by which we rotate the fragment along the bond the two fragments are attached to each other and arrive at a new state s_1. The initial pose of the new fragment, before the agent picks the torsion angle, is given by a 'pre-torsion embedding' as shown in Figure 1, which we predict using a second SE(3) Transformer. This pre-torsion embedding is trained using expert data once, in the beginning of the framework, to predict the correct pose up to the symmetry group of the fragment and remains fixed for the training of the actual torsion policy. Moreover, we encode the entire 2D topological molecular graph (without 3D coordinates) using a vanilla graph convolution network Xu et al. [2018], as well as the 3D position of the atoms that have already been placed and concatenate the output of this network to the input features of s for the SE(3)-Transformer policy. Compared to prior approaches that choose all angles simultaneously, such as Gogineni et al. [2020], we believe our autoregressive approach has a clearer definition of the problem given that the required number of steps in the environment is known a priori. This prevents the agent from "walking in cycles" created by revisiting states from prior actions.

2.3 Molecular Symmetries

Previous work has shown that it is beneficial for molecular property prediction tasks to incorporate inductive biases about the rotation and translation symmetries of a molecule into the neural network

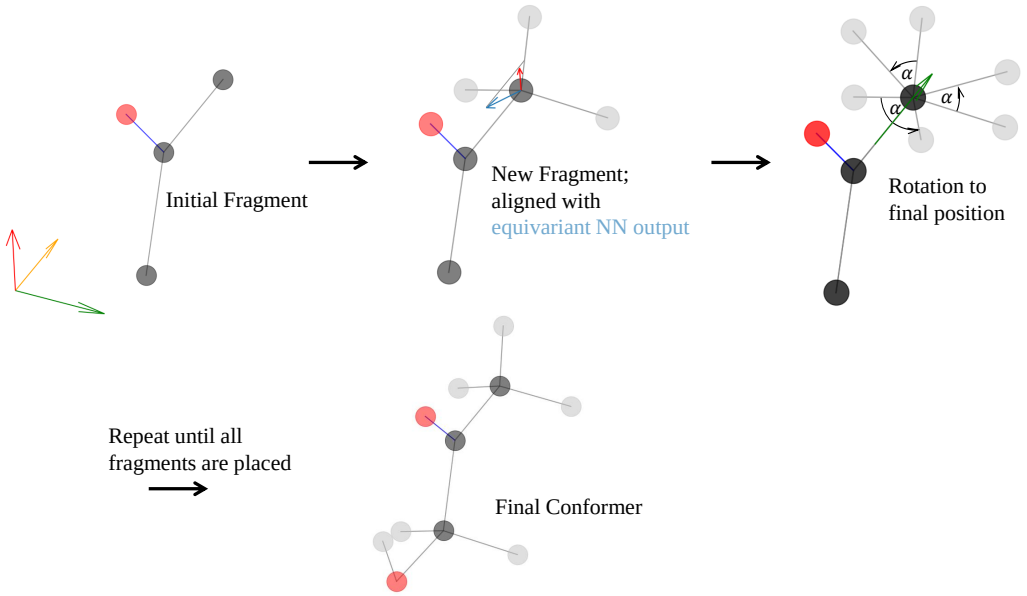

Figure 1: Workflow of the agent: We start with a random fragment from the query graph. Since we work in the rigid-rotor approximation, we know angles and bond distances, allowing us to place the fragment in space. Next, we input this fragment to our neural network to predict a rotation equivariant output vector. We use this to align the next fragment to this vector, yielding a rotation equivariant embedding. The resulting molecular graph is then fed into another neural network to predict a rotation invariant angle α, which indicates how much we need to rotate the fragment to get the final position. We repeat this process until all fragments are placed.

architecture. We achieve this by constraining the architecture in such a way that all features are rotation equivariant. A function $f(x) : V \to Y$ is equivariant with respect to some symmetry group G, if for every transformation $T_g : V \to V$ and $S_g : Y \to Y$ with $g \in G$, the function f transforms as $f(T_g x) = S_g f(x)$. The SE(3)-Transformer architecture [Fuchs et al., 2020] we use represents data in the spherical harmonics basis and can produce outputs of different "types" for each node and edge. The "0-type" output is a rotation invariant scalar, while the "1-type" output is a rotation equivariant three-dimensional vector. This allows us to construct a rotation invariant agent to make efficient use of the available data during training. Specifically, we want a given molecule τ and its rotated and translated version to have the same probability under the policy. Therefore, for a rotation matrix $R \in \mathbb{R}^{3\times3}$ and $t \in \mathbb{R}^3$, $\pi_\theta(\tau) = \pi_\theta(R\tau + t)$ should for every $R\tau + t$ along a rotated or translated trajectory. One way of achieving this would be to predict the dihedral angle between one of the atoms in the new fragment and one of the atoms in the old fragment. However, the choice of the two reference atoms is not always unique. Instead, we perform a two-step process: First, we equivariantly embed the new fragment in some initial position in space as show in Figure 1. At this step, the agent can predict the rotation up to the symmetry of the fragment given that there is no canonical way to choose the attachment point of the new fragment. The rotation of the fragment by α_{inv} is predicted using a rotation invariant radial distribution.

2.4 Radial Distribution

Angles possess a circular topology since 0 and 2π represent the same angle. As such, we model our action distribution as a circular distribution described by a circular probability density function. Specifically, we apply the projected normal distribution [Hernandez-Stumpfhauser et al., 2016], which is described by a mean and a variance parameter. To increase the model's flexibility and allow for multi-modality, we use a "mixture of projected normals" distribution with K mixture components. We parameterize the direction of the distribution using K rotation equivariant output vectors from the SE(3)-Transformer and project them onto the plane defined by the torsion axis (shown by the red arrow in Figure 1). The length of the same vectors parameterizes the variance in the distribution. While we desire our final distribution to be rotation invariant, the output from the current formulation

is rotation equivariant. In order to make our output rotation invariant, we predict one more equivariant output in the same way we described above and use it as a measuring reference for the angles. Since two equivariant vectors maintain their relative angle under rotations, the resulting distribution is rotation invariant. Figure 2 depicts some of those distributions on a set of torsion actions.

3 Preliminary Results

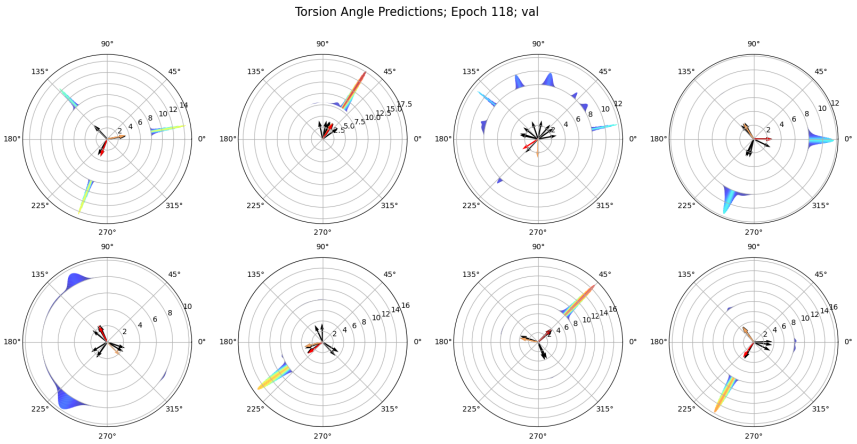

Figure 2: A random batch of action distributions on the validation set. Black Arrows: The mixture component of Mixture of Projected Normal distribution. Light red arrow: Reference vector, with respect to which angles are measured. Red arrow: The target, rotated in the coordinate system of the reference vector. Overall, we can see that the model is converging to the desired with higher confidence as expressed by the peaky distributions in the image.

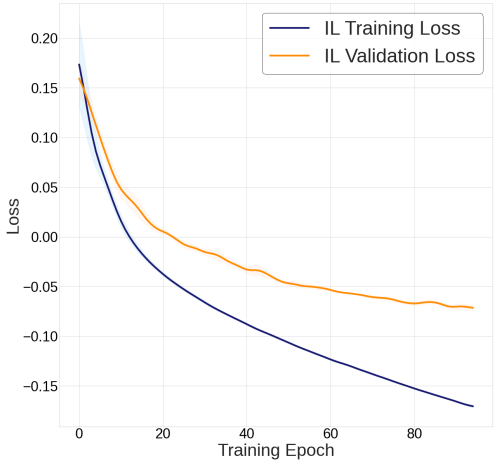

Figure 3: Behavioural Cloning Training and Validation Curves for QM9-X

We conducted experiments with our behavioural cloning formulation described in Section 2 for a subset of molecules from QM9 that do not include ring structures, which we name QM9-X. The exclusion of ring structures was primarily due to time constraints with the aim being to include ring structures in future work. We split QM-X into a training and a validation set and subsequently decomposed the molecules into states and actions that led to the ground truth conformer, resulting in 269130 training and 95360 validation samples. We first train the neural network encompassing the "pre-torsion" embedding and apply it for training of the torsion policy using the QM9-X data as expert policy trajectories.

From Figure 3, we see that the BC training is stable and with low variance. To get a better insight into the model, in Figure 2 we visualize the predictions of a random batch in the validation set. The black arrows represent the mixture components, the light red arrow is the reference vector, and the dark red vector is the target angle, rotated into the coordinate system of the reference vector. We see that most predictions successfully put most of the probability mass around the target vector. The third plot in the top row represents a linear fragment, where no dihedral angle can be defined and can therefore be ignored. The leftmost plot in the top row corresponds to a tetrahedral fragment with three hydrogen atoms, which has equal probability in all three directions separated by 120 degrees correctly determined by the policy. We also some ambiguous cases where the model cannot distinguish between two directions that are not entirely symmetric. We aim to resolve those cases,

and to further improve performance of the policy, with subsequent fine-tuning using energy-based rewards in a reinforcement setting. Overall, the imitation learning results appear very encouraging as a starting point for subsequent experiments centering on deep reinforcement learning for further improvements.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

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*, 34:13757–13769, 2021.

Tarun Gogineni, Ziping Xu, Exequiel Punzalan, Runxuan Jiang, Joshua Kammeraad, Ambuj Tewari, and Paul Zimmerman. Torsionnet: A reinforcement learning approach to sequential conformer search. *Advances in Neural Information Processing Systems*, 33:20142–20153, 2020.

Daniel Hernandez-Stumpfhauser, F. Breidt, and Mark Woerd. The general projected normal distribution of arbitrary dimension: Modeling and bayesian inference. *Bayesian Analysis*, 12, 01 2016. doi: 10.1214/15-BA989.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

Greg Landrum. Rdkit: Open-source cheminformatics. URL `http://www.rdkit.org`.

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

Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):1–14, 2011.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

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*, pages 11537–11547. PMLR, 2021.