RL Boltzmann Generators for Conformer Generation in Data-Sparse Environments

Yash Patel[1], Ambuj Tewari[1]
[1]Department of Statistics, University of Michigan

UNIVERSITY OF MICHIGAN

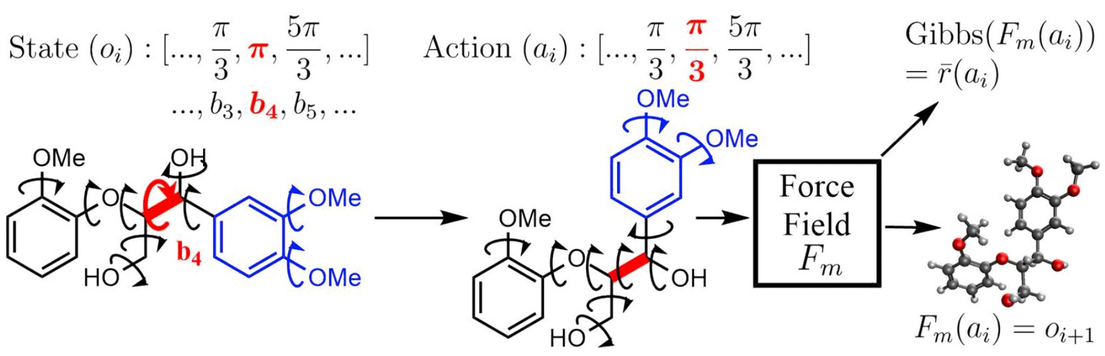
Introduction

Many proteins of the human proteome take on multiple shapes, or "conformations." An extreme example of these are "intrinsically disordered proteins" (IDPs), which are not captured by current state-of-the-art protein folding solutions. Relatedly, recent work has investigated using machine learning for conformer generation. Despite performing well on small molecules, these models all suffer from the drawback that they require substantial training data. Such data ultimately come from MD simulations or experiments, both of which currently fail for IDPs.

A potential work-around that has been explored in parallel is to train directly against the unnormalized Boltzmann distribution, explored by a subset of generative models known as "Boltzmann generators." However, training solely against the energy function has been observed to result in concentrated sampling on stable conformers. This finding, however, has only been replicated in the context of normalizing flows. It is, therefore, difficult to disentangle whether such mode collapse is a function of the modeling technique or is more fundamentally linked to the isolated training against the energy.

Our main contribution, therefore, is to demonstrate that training an RL agent with a "Gibbs reward," which closely parallels training directly against the energy, also exhibit symptoms of mode collapse, suggesting that this issue is more fundamentally linked to the isolated use of the energy function than to the modeling modality.

Statistical Model



We fix a molecule of interest. States S and actions A are defined over the dihedral angles of the molecule backbone: $s, a \in [0, 2\pi]^N$ where N is the number of dihedral angles. Actions are further discretized to be multiples of $2\pi/M$ with M = 6 to improve training.

Dynamics through the environment evolve through sampling of the action space from a policy $a_t \sim \pi(s_{t-1})$. A temporary state is produced by directly acting upon state s_{t-1} with the torsion angles a_t. This state is then relaxed using a force field F, specifically MMFF, to produce the next state s_t. This procedure is then repeated T times, ultimately producing a sequence of conformers $s := \{s_0, s_1, s_2, ..., s_T\}$.

To reward the agent, a "Gibbs Score" is initially formulated as follows, with $U(x)$ computed using the same classical force field used for relaxation and where Z_0 and U_0 normalization factors:

$$\text{Gibbs}(s_t) = \frac{1}{Z_0} \exp\{-(U(s_t) - U_0)/k\tau\}$$

To avoid encouraging repeated sampling, we prune similar states before computing the reward. Denoting the current "history" of configurations the molecule has been in as $s := \{s_i\}$, the current conformer is pruned if it is similar to one that has been previously observed as measured by the Torsion Fingerprint Distance (TFD):

$$\exists\, k \text{ s.t. } D_{\text{TFD}}(s_t, s_k) \leq \epsilon$$

Agent Model

The policy π consists of a Graph Neural Network trained iteratively using PPO. π is an edge-network MPNN:

$$x_i^{t+1} = \Theta x_i^t + \sum_{j \in \mathcal{N}(i)} h(x_j^t, e_{i,j})$$

where $\{x_i\}$ are the embeddings for the N atoms in the molecule, $e_{i,j}$ is the edge information between atoms (i, j), Θ is a GRU, and h is an MLP. After each iteration t, the output embeddings are aggregated using set-to-set pooling, which gives us an aggregated embedding of the overall molecule y_t. The history of molecular embeddings $\{y_i\}$ are passed through an LSTM to give a current "aggregated" state g_t.

To finally produce the actions, for each torsion T_i, we compute p_i^t and then sample actions $a_i^t \sim p_i^t$:

$$T_i = (b_1^i, b_2^i, b_3^i, b_4^i) \qquad p_i^t := f(x_{b_1^i}, x_{b_2^i}, x_{b_3^i}, x_{b_4^i}, g^t)$$

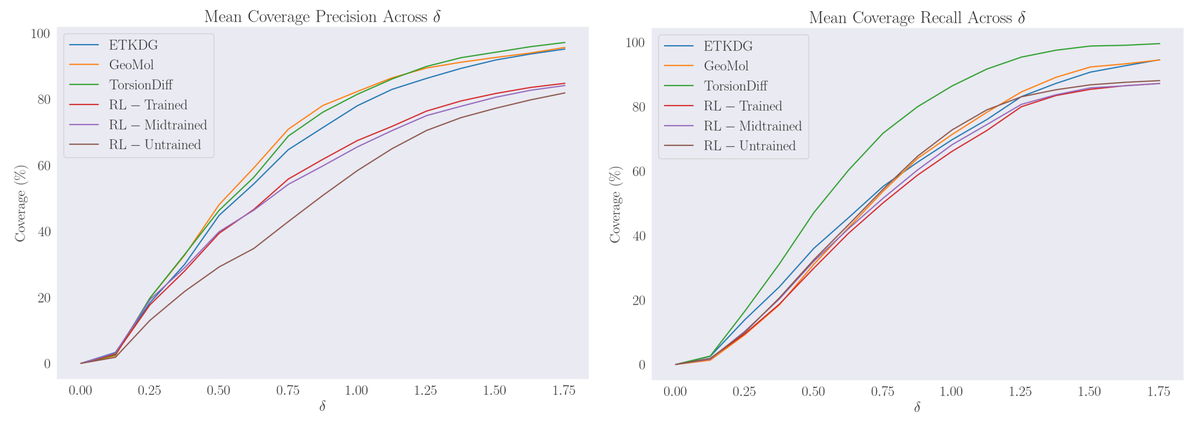
Results

Experiments with GEOM-Drugs were conducted on a randomly sampled subset of 100 molecules using the following metric:

$$\text{COV-R}(S_g, S_r) = \frac{1}{|S_g|} \left| \{\mathcal{C} \in S_r \mid \text{RMSD}(\mathcal{C}, \widehat{\mathcal{C}}) \leq \delta, \widehat{\mathcal{C}} \in S_g\} \right|$$

S_r, S_g represent the reference and generated conformer ensembles. δ is an arbitrary threshold used for designated two conformers as being "equivalent" for the purposes of assessing discovery. The precision metric can be defined simply by exchanging S_g and S_r.

The RL agent performance, for both metrics, is plotted at three stages of the training: at step 0, step 50,000, and step 100,000, respectively referred to as "untrained," "midtrained,", and "trained.



Note that the recall of the RL agent decreases over the training run in exchange for its precision increasing, respectively seen in the rightward shift of the recall curves and upward shifts of the precision curves for progressively more trained RL agents. This implies the RL agent conformer generation becomes more highly concentrated on a smaller subset of the conformer space.

Conclusion

A current line of investigation we are pursuing is motivated by the curriculum learning: given that MD simulations *can* be tractably run on subsets of large proteins, such data can be used as a curriculum in an RL Boltzmann generator. That is, for a protein of interest, progressively longer subsequences of its amino acid chain could be taken, some of which could be feasibly analyzed with MD. For such subsequences, the Boltzmann generator could be trained with both the energy and the generated MD ground truths, with the intention being that subsequences where no such truth is available would begin from an informed "prior" for energy-based sampling.