Geometric Constraints from Molecular Dynamics to Graph Generative Modelling

Anonymous Authors[1]

Abstract

Macroscopic properties of molecules and protein complexes are described by statistical ensembles of the systems i.e. the binding free energy, which are generally intractable to compute making integral over this space in need of good approximations. Generative models particularly have found great potential in obtaining tractable densities and samples. In order to study interesting partitions of the state-space it becomes desireable to sample states that satisfy constraints either because we have prior structural knowledge or only certain subsets of the state-space are of interest. Here we propose a method that allows one to generate samples that satisfy any number of geometric constraints in Euclidean spaces, i.e. distances, torsions, or dihedrals, by integrating a constraint projection operator into the formalism of Denoising Diffusion Probabilistic Models.

1. Introduction

Infinitesimal Dynamics in classical mechanics is commonly formalized by lagrangians. By solving for functionals that extremize the lagrangian one obtains equations of motion. In molecular systems, e.g. Molecular Dynamics, the EOM are: $M\frac{d^2x}{dt^2} = -\nabla U - \sum_a \lambda_a \nabla\sigma_a$, where M is the diagonal mass matrix, x the cartesian coordinates, t is time, and U is the potential energy. The σ_a are a set of holonomic constraints and λ_a are the Lagrange multiplier coefficients. To generalize from holonomic to nonholonomic constraints, one can use slack variables to transform the latter into the first. For example, we can add a slack variable $y \geq 0$ and define d_j as the boundary of a nonholonomic constraint. Then, we can express the constraint as:

$$\sigma_a := ||x_{aj} - x_{ak}||_2^2 - d_j \leq 0 \rightarrow ||x_{aj} - x_{ak}||_2^2 - d_j + y = 0.$$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.

Starting with $z_x, z_h = f(x,h) = [x(0), h(0)] + \int_0^1 \phi(x(t), h(t))dt$ with z being a latent vector sampled from gaussians and the indexes x and h indicate the latent variables associated to the coordinates of each particle and the vector embedding of each particle, ϕ is the parameterized transformation defined by a equivariant graph neural network. This defines a Neural ODE [Che+18] which generalizes to Denoising Diffusion Probabilistic Models [HJA20]. This form of transformation has the same infinitesimal nature as our previous EOM which makes it acceptable to apply sets of constraints via Langrange's Multipliers, analogous to solving our EOM and thus one can insure the continual satisfaction of a set of constraints using the SHAKE algorithm from Molecular Dynamics.

In the following, we will give a summary of the SHAKE algorithm and segments of the equivariant normalaizing flow necessary to elaborate on how to combine them. Next, it will be elaborated that the spaces of latent embeddings and output samples are generally of very different nature, and constraints defined in one space will not necessarily be useful in the other. We suggest a continuous transformation of the constraints such that they are always satisfied in the latent space, and become more restrictive throughout the integration. We leave to further work the generalization of parameterized continuous transformations of the constraint set.

2. Previous Research

Generative models of molecules and proteins have been a subject of interest in recent years. A number of different approaches have been proposed in the literature. [HN19] introduced a method for generating valid Euclidean distance matrices. This method is important for the generation of molecules, as it ensures the resulting molecular structures are physically realistic. In the work by [Noé+19], Boltzmann Generators were proposed to sample equilibrium states of many-body systems with deep learning. This method is particularly useful for generating molecular configurations that obey the laws of thermodynamics.

[SHW21] proposed Equivariant Graph Neural Networks, which can be applied to model molecules and proteins. The equivariance property of these networks ensures that their predictions are consistent under different orientations and permutations of the molecule.[Hoo+23] further extended the concept of equivariant networks to the diffusion process for 3D molecule generation. Their method maintains the advantages of equivariance, while allowing more flexibility in the generation process. [Cor+23] applied similar modelling techniques to diffusion models on protein ligand complexes. Lastly, [Jin+23] devise a method of protein generation models that diffuse over harmonic potentials.

The shake algorithm, described in a parallelized fashion by [ERH11], enforces constraints on molecular dynamics simulations of chemicals and biomolecules. This algorithm is conventionally used in simulations to get rid of high frequency motions, i.e. those seen in bonds between atoms. By incorporating the shake algorithm, our constraint denoising diffusion method effectively models more complex constraint sets. These works together provide a solid foundation for the development of generative models for molecules and proteins. They highlight the importance of incorporating physical principles and mathematical structures into these models.

3. Constrained Generative Processes

3.1. Geometric Constraints in Shake

First, we define the constraint functions for the pairwise distance (not necessarily between bonded atoms), bond angle, and dihedral angle. We can additionally create nonholonomic constraints via slack variables as described below.

$$\sigma_{d_{ij}} = (d_{ij} - d_{ij,0})^2 = 0 \tag{1}$$

$$\sigma_{\theta_{ijk}} = (\theta_{ijk} - \theta_{ijk,0})^2 = 0 \tag{2}$$

$$\sigma_{\psi_{ijkl}} = (\psi_{ijkl} - \psi_{ijkl,0})^2 = 0 \tag{3}$$

These constraint functions compare the current pairwise distance, bond angle, and dihedral angle with their target values, and the goal is to minimize the difference.

Next, modify the constraint matrix in the SHAKE algorithm to include pairwise distance, bond angle, and dihedral angle constraints seen in equation 4, where ij, ijk, and $ijkl$ sum over the pairwise, bond angles, and torsion constraints indicating the number of atoms in each type of constraint type. The constraint matrix now accounts for the pairwise distance, bond angle, and dihedral angle constraints by including their second-order derivatives with respect to the Cartesian coordinates by including their contributions to the Lagrange multipliers. After solving for the Lagrange multipliers, update the coordinates using the adjusted coordinate set equation like before. It is also possible to try to optimize the coordinates via other optimization algorithms like ADAM or SGD.

In this section, we discuss the methods needed to understand how constraints can be represented, and define a novel diffusion process which projects the dynamics onto the submanifold defined by arbitrary sets of geometric constraints.

3.2. Shake Algorithm

The SHAKE algorithm takes as input a set of coordinates x of a molecular system and a set of constraints σ. At each time step the coordinates are updated according to the equations of motion (EOM) at hand (without constraint terms) and subsequently are corrected. In general, the EOM will lead to dynamics that do not satisfy the constraints, and thus this correction is mandatory.

Assuming masses of all the particles and delta time are unit we have the following equation for updating x_i iteratively until the constraints are satisfied.

$$x_i^{(n)} = x_i^{(n-1)} - \sum_b \lambda_b^{(n-1)} \nabla\sigma_b(x_i) \tag{5}$$

where $x_i^{(n)}$ is the updated coordinate after n iterations of satisfying constraints at each time step, x_i is the initial coordinates at each time step, and $\lambda_b^{(n-1)}$ is the lagrange multiplier for each constraint σ_a. The equation to solve at each iteration of each time step is

$$\sum_\beta \lambda_\beta^{(n-1)} A_{\alpha\beta}^{(n-1)} = \sigma_\alpha(x_i^{(n-1)}) \tag{6}$$

with

$$A_{\alpha\beta}^{(n-1)} = \nabla\sigma_\alpha(x_i^{(n-1)})\nabla\sigma_\beta(x_i). \tag{7}$$

The matrix $A_{\alpha\beta}^{(n-1)}$ is a symmetric matrix that describes how changes in particle positions affect both potential energy and constraint violations. The elements of the matrix are given by:

$$A_{\alpha\beta}^{(n-1)} = \frac{\partial^2 U}{\partial x_\alpha \partial x_\beta} + \sum_{k=1}^{N_c} \lambda_k^{(n-1)} \frac{\partial^2 \sigma_k}{\partial x_\alpha \partial x_\beta} \tag{8}$$

where N_c is the number of constraints. The matrix $A_{\alpha\beta}^{(n-1)}$ is used to solve for the Lagrange multipliers $\lambda_\beta^{(n)}$, which are then used to adjust particle positions.

3.3. Constraint-Induced Diffusion Process

Suppose we want to incorporate a constraint, such as a distance constraint between two atoms. Let's denote this

$$A^{(n-1)}_{\alpha\beta} = \frac{\partial^2 U}{\partial x_\alpha \partial x_\beta} + \sum_{ij} \lambda^{(n-1)} d_{ij} \frac{\partial^2 \sigma_{d_{ij}}}{\partial x_\alpha \partial x_\beta} + \sum_{ijk} \lambda^{(n-1)} \theta_{ijk} \frac{\partial^2 \sigma_{\theta_{ijk}}}{\partial x_\alpha \partial x_\beta} + \sum_{ijkl} \lambda^{(n-1)} \psi_{ijkl} \frac{\partial^2 \sigma_{\psi_{ijkl}}}{\partial x_\alpha \partial x_\beta} \tag{4}$$

constraint by $f(x) = 0$ for simplicity. We can modify the diffusion process to satisfy this constraint by projecting the noise term onto the nullspace of the gradient of the constraint function, analagous to the A matrix in SHAKE. This gives us:

$$dx = \sqrt{2D}(I - \nabla f(x)(\nabla f(x))^T)dB - D\nabla \log p_t(x)dt$$

where D is the diffusion constant, B is a standard Brownian motion, and $\nabla \log p_t(x)$ is the gradient of the log-probability density, which is equivalent to the negative of the potential energy function of the system. Here, I is the identity matrix, and $\nabla f(x)(\nabla f(x))^T$ is the outer product of the gradient of the constraint function, which represents the direction in which the constraint is changing. This projection ensures that the noise term does not push the system out of the constraint-satisfying space.

The covariance matrix of the perturbed Gaussian distribution of the denoising process can be understood formally using the Schur complement method, not discussed for space. The key takeaway is the relation between constraints and correlations via projecting out the constraints in the Covariance matrix of a Multivariate Gaussian. This modified covariance matrix then defines the perturbed Gaussian distribution from which we can sample at each time step of the diffusion process. This is a good approximation when the constraints are nearly linear or when the changes in the variables are small. One note is that in general, the order of projection and sampling does matter, but since we deal with linearized constraints or small changes this is negligible as seen in the original SHAKE formalism.

3.4. Nonholonomic Constraints

We are more interested in nonholonomic constraints where each constraint has possibly a lower and upper bound. As we mentioned earlier, by adding a slack variable one can translate the nonholonomic constraints to holonomic ones. To formalize this, one sees that a constraint having a lower and upper bound will either be completely satisfied or fail to satisfy a single boundary. Thus, we only have to consider at most one holonomic constraint at each call to $SHAKE$ meaning each constraint with a lower and upper bound may be replaced by a lower, upper, or no bound for each call.

To calculate the slack variable y from $\sigma_{jk} := \|x^l_i - x^l_j\| - d_{jk}$ which is $\leq or \geq 0$, one has

$$y = \begin{cases} max(0, ||x^l_i - x^l_j|| - d^u_{jk}), & \text{if } \leq \\ max(0, d^l_{jk} - ||x^l_i - x^l_j||), & \text{if } \geq \end{cases} \tag{9}$$

Algorithm 1 Pseudo-Code for Training

```
$t \sim U(0,T)$, $\epsilon \sim N(0,I)$
Subtract center of gravity from $\epsilon$: $\hat{\epsilon} = [\epsilon(x), 0] - [x, 0]$
Compute $z_t = \alpha_t[x,h] + \sigma_t\hat{\epsilon}$
Update $z_t \to x + \epsilon_s$, where $\epsilon_s = \text{shake}(z_t) - \alpha_t x$
Compute $\epsilon'_s = \text{shake}(\varphi(z_t) + z_t) - z_t$
Minimize $\mathcal{L}_c = |\epsilon_s - \epsilon'_s|^2_2$
```

where d_{jk} is the lower or upper bound in case of nonholonomic constriants and the defined constraint value for holonomic constraints.

In the generative process, we define the initial values of d_{jk} such that the constraints have little effects. The constraints are then linearly interpolated throughout the ODE until the predetermined boundary values of d_{jk} are reached.

3.5. Training and Sampling Algorithms

3.5.1. Training Process

During training, in Algorithm 2, we first sample a time step t and noise vector ϵ from uniform and Gaussian distributions respectively. Then subtract the center of gravity from the noise vector to ensure that it lies on a zero center of gravity subspace. Then compute the latent variable z_t by scaling and adding the input coordinates $[x, h]$ with the noise vector. Finally, minimize the difference between the estimated noise vector and output of the neural network to optimize EDM. For each molecule between 5 and 15 constraints are sampled from x for each batch element. The constraints are uniformly sampled from the pairs, triples, and quadruplets of the atom set of each molecule. This adds an extra layer of complexity due to the constraint distribution which we need to sample from the true data distribution.

3.5.2. Generative Process

In this algorithm, first sample a latent variable z_T from a Gaussian distribution. Then iterate backwards through time and sample noise vectors ϵ at each step. Subtract the center of gravity from the noise vector to ensure that it lies on a zero center of gravity subspace. Then compute the latent variable z_s by scaling and adding the input coordinates with the noise vector and previous latent variable. Finally, sample the input coordinates $[x, h]$ from a conditional distribution given the initial latent variable z_0. The shake algorithm enforces the constraints, as in training, at each sampling step during generation.

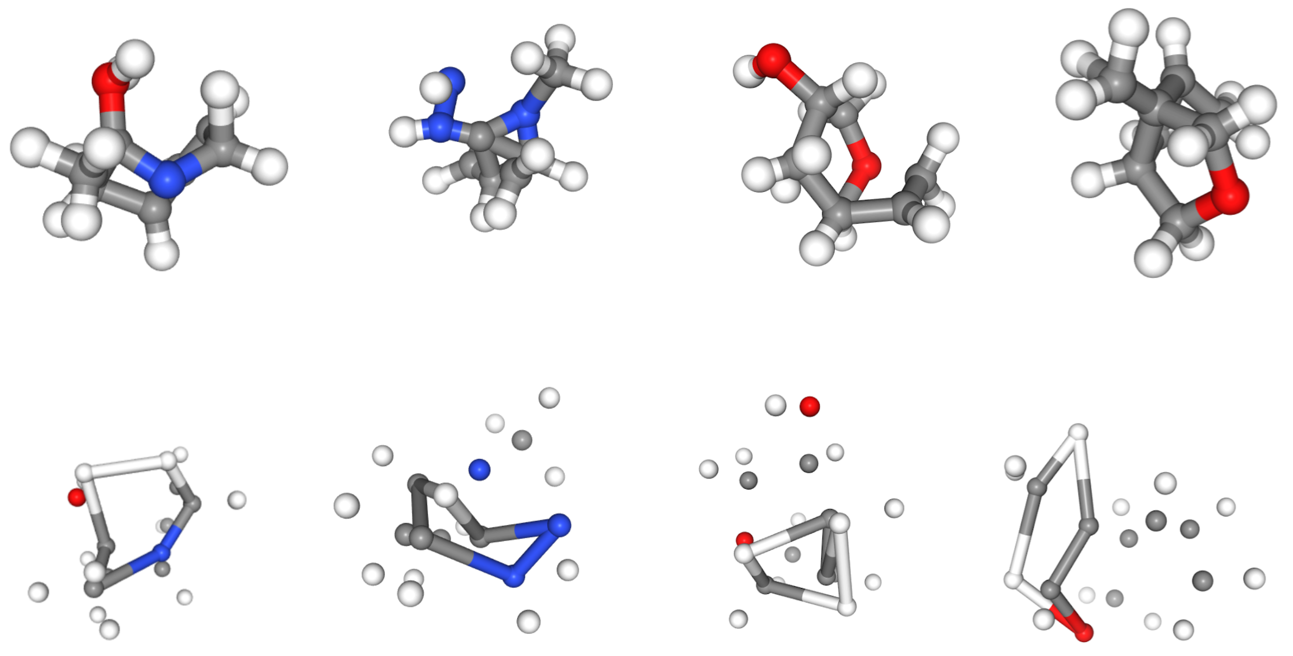

Figure 1: Molecules generated with 6 atom cyclic constraints between 1.3-1.5 Angstroms each with bounds of .1 Angstrom. Atom types are generated as well, so we can not arbitrarily encode constraints between specific types of atoms in our current implementation, but this will be possible in further developments.

4. Experiments

In the experimental section of our study, we evaluate our proposed method by generating molecules with cyclic constraints in Figure 1. The cyclic constraints impose specific geometric relationships among atoms in a molecule, such as the bond distances, bond angles, and torsional angles, which are essential for maintaining the chemical stability and physical plausibility of the generated molecules.

During the training phase, constraints are sampled from the dataset. This approach encourages the model to learn the distribution of constraints inherent in the training data, which reduces the Kullback-Leibler (KL) divergence between the data distribution and the model distribution. Consequently, the KL divergence during training is always minimized, promoting the model to generate molecules that closely resemble those in the training set.

For the practical implementation of this training procedure, we began with a pre-trained model provided by Welling et al.Our methodology then fine-tuned this pre-existing model using our constraint projection method. Due to time considerations and simplicity, our training and experiments focused on molecules consisting of 21 atoms.

5. Discussion

Our method serves as a potent tool for incorporating complex constraints in denoising diffusion processes, specifically when dealing with multi-constraint specifications. Its iterative nature allows it to address nonlinear constraint problems and extends the power of denoising diffusion probabilistic models to work with constraints. Thus allowing these models to leverage the structure inherent in many physical systems. Indeed, many of these systems come with prior structural knowledge, including geometric information like distances, torsions, bond angles, and generalizeable to other piece-wise polynomial terms. Such information can significantly enhance the training process and enable explicit sampling of subsets of the state space.

Although constraints can guide generation towards more physically plausible structures, there can be potential instability in the generation process. This instability may originate from discrepancies between constraints used during training and those applied during generation. It underlines the need for further work to establish robust training procedures that align more closely with the generation constraints. Especially, with application focused studies like generating peptides or ligands with specific interaction profiles with a given protein.

References

[ERH11] R. Elber, A. P. Ruymgaart, and B. Hess. “SHAKE Parallelization”. In: *The European Physical Journal Special Topics* 200 (2011), pp. 211–223.

[Che+18] Ricky T. Q. Chen et al. “Neural Ordinary Differential Equations”. In: *arXiv preprint arXiv:1806.07366* (2018). URL: https://arxiv.org/abs/1806.07366.

[HN19] M Hoffmann and F Noé. “Generating Valid Euclidean Distance Matrices”. In: *arXiv preprint arXiv:1910.03131* (2019). URL: https://arxiv.org/abs/1910.03131.

[Noé+19] Frank Noé et al. “Boltzmann Generators: Sampling Equilibrium States of Many-Body Systems with Deep Learning”. In: *Science* 365 (6457 2019). ISSN: 0036-8075. DOI: 10.1126/science.aaw1147.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *arXiv preprint arXiv:2006.11239* (2020). URL: https://arxiv.org/abs/2006.11239.

[SHW21] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E (n) Equivariant Graph Neural Networks”. In: *International Conference on Machine Learning*. 2021, pp. 9323–9332.

[Cor+23] Gabriele Corso et al. “DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking”. In: *International Conference on Learning Representations*. 2023. URL: https://arxiv.org/abs/2210.01776.

[Hoo+23] Emiel Hoogeboom et al. “Equivariant Diffusion for Molecule Generation in 3D”. In: *International Conference on Machine Learning*. 2023, pp. 8867–8887. URL: https://arxiv.org/pdf/2203.17003.pdf.

[Jin+23] Bowen Jing et al. “EigenFold: Generative Protein Structure Prediction with Diffusion Models”. In: *International Conference on Learning Representations, Machine Learning and Data Driven Discovery workshop*. 2023. URL: https://arxiv.org/abs/2304.02198.