Constrained Sampling of Discrete Geometric Manifolds using Denoising Diffusion Probabilistic Models

Justin Diamond [1]
Markus Lill [1]

Abstract

Understanding the macroscopic characteristics of biological complexes demands precision and specificity in statistical ensemble modeling. One of the primary challenges in this domain lies in sampling from particular discrete subsets of the state-space, driven either by existing structural knowledge or specific areas of interest within the state-space. We propose a method that enables sampling from distributions that rigorously adhere to arbitrary sets of geometric constraints in Euclidean spaces. This is achieved by integrating a constraint projection operator within the well-regarded architecture of Denoising Diffusion Probabilistic Models, a framework founded in generative modeling and probabilistic inference. The significance of this work becomes apparent, for instance, in the context of deep learning-based drug design, where it is imperative to sample from the discrete structures of the solution space.

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.

Neural ODE [Che+18] which generalizes to Denoising Diffusion Probabilistic Models [HJA20] have 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 a generalization of the SHAKE algorithm from Molecular Dynamics.

The problem we hope to model are non-linear constraints where constrained subsets of atoms determine the unconstrained subset to a high degree. We argue these types of non-linear constraints are important in the field of generative drug development where generated molecules must satisfy certain structural or analytic properties a priori. Additionally, we not that the solution space of physically meaningful configurations can be modelled as a discrete distribution. We are considering a situation where the underlying discrete distribution is unknown, and we aim to sample from it based on sets of progressively refined constraints defining discrete manifolds.

Given a discrete space X, partitioned into topologically disjoint subsets (regions) $X_1, X_2, ..., X_k$ by progressively added or refined constraints, or the multi-valued solution set of the constraint set. It is difficult to directly sample from X due to the discrete nature of the space. However, denoising diffusion probabilistic models (DDPMs) can aid in this situation by modeling the discrete distribution as a diffusion process.

Consider a Markov chain $Z_0, Z_1, ..., Z_T$ where Z_T is distributed according to the desired discrete distribution on X (which we wish to sample from) and Z_0 is an easily sampled noise distribution (e.g., uniform or Gaussian). The DDPM defines a reverse process where Z_t is obtained by denoising Z_{t+1}, modeled as $Z_t = g_t(Z_{t+1}, \varepsilon_t)$ for each $t = T-1, ..., 0$, where g_t are deterministic denoising functions and ε_t are independent noise variables. Training involves learning the g_t's from data. To sample, we start from Z_0 and apply the sequence of learned constrained-denoising functions to obtain a sample from the desired discrete distribution on X.

In the following, we will give a summary of the SHAKE algorithm and extend it to our setting. 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 continue by describing the formal definition of such a constraint projection method and show simple examples where complex

constraints are satisfied within small molecules. We highlight areas of continued research to better understand the discrete solution space. We leave to future work the study of this methodology to larger systems, and more application based studies.

2. Previous Research

Generative models of graphs have been a subject of interest in recent years. A number of different approaches have been proposed in the literature. [HN19] generates valid Euclidean distance matrices ensuring the resulting molecular structures are physically realistic which are then reconstructed in 3D space. In [Noé+19], Boltzmann Generators sample equilibrium states of many-body systems with deep learning, useful for generating molecular configurations that thermodynamics distributions.

[SHW21] proposed Equivariant Graph Neural Networks, which can be applied to model molecules and proteins while ensuring that their predictions are consistent under different orientations and permutations of the molecule.[Hoo+23] further extended the concept to the diffusion process for 3D molecule generation. [Cor+23] applied similar methodologies to diffusion models on protein ligand complexes, and [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 linear 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.

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.

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

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

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

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. We can additionally create nonholonomic constraints via slack variables. 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:

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

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:

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

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

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 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, available in the Appendix. 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.

4. Disjoint Solution Space as Discrete Structures

4.1. Formalizing Disjoint regions

Definition 1: Discrete Approximation

Let M be a solution set in $\mathbb{R}^n$ defined by a single set of constraints. A discrete approximation of M, denoted as M', is a finite subset of M that satisfies the following conditions:

Covering Condition: For every point $p \in M$, there exists a point $p' \in M'$ such that the Euclidean distance between p and p' is less than or equal to a resolution parameter $\epsilon > 0$. Mathematically, $\forall p \in M, \exists p' \in M' : |p - p'| \leq \epsilon$.

Connectivity Condition: Any two points $a, b \in M'$ can be connected by a sequence of points in M', with each pair of consecutive points being at most distance ϵ apart. Formally, $\forall a, b \in M', \exists p_1, p_2, \ldots, p_k \in M'$ such that $p_1 = a$, $p_k = b$, and $|p_i - p_{i+1}| \leq \epsilon$ for $i = 1, 2, \ldots, k-1$.

The discrete approximation M' is a discretized version of M that forms one or more finite connected components, each representing a distinct solution to the given set of constraints up to the resolution ϵ.

Definition 2: Topological Disconnected Regions

Given a topological space X and a subset M' of X, we say that two subsets M_i' and M_j' of M' are topologically disconnected if there exists no continuous path in M' from a point in M_i' to a point in M_j'.

The term "continuous path" refers to a sequence of points in M', denoted by $p_1, p_2, \ldots, p_k$, such that p_1 belongs to M_i', p_k belongs to M_j', and $|p_i - p_{i+1}| \leq \epsilon$ for $i = 1, 2, \ldots, k-1$.

In the context of a single set of constraints that may lead to multiple solutions, the discrete approximation M' will consist of multiple connected components, each corresponding to a distinct solution. Each component satisfies the covering condition and the connectivity condition within itself, but there is no requirement for a continuous path between different components. Therefore, each of these components can be considered a topologically disconnected region.

4.2. Chirality and Disjoint Regions

Alanine can exist in two chiral forms or "enantiomers", commonly referred to as D-alanine and L-alanine. These forms are mirror images of each other. Even though they share the same set of constraints (same atoms, same bonds, same bond angles), these molecules are not the same due to their different spatial arrangements. The D-alanine and L-alanine molecules represent two different solutions to the same set of constraints, and we can consider them as two distinct "topologically disconnected regions" in the solution space.

The two regions are disconnected because there's no way to "morph" a D-alanine molecule into an L-alanine molecule

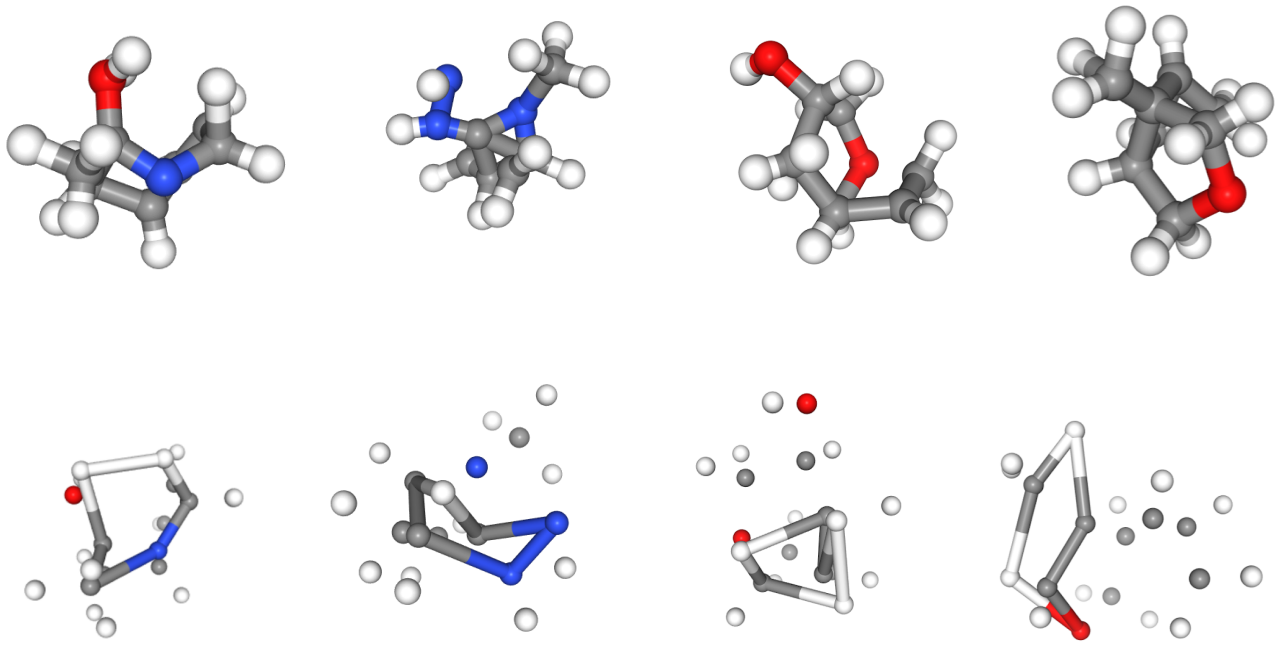

Figure 1: Molecules generated with 6 atom cyclic constraints between 1.3-1.5 Angstroms each with bounds of .1 Angstrom. Atom types are generated as well, so we can not arbitrarily encode constraints between specific types of atoms in our current implementation, but this will be possible in further developments.

by a continuous deformation (i.e., without breaking and reforming bonds). In the solution space, this implies that there's no path from a point in the D-alanine region to a point in the L-alanine region without crossing the boundary between the regions. Thus we say they are topologically distinct and thus form separate regions in the solution space.

In general, adding of new constraints, varying existing constraints, and considering a single set of constraints that has multiple distinct solutions can lead to topologically disconnected regions in the solution space. These disjoint regions are combinatoric and require new tools to be studied accurately.

5. Experiments

We illustrate our proposed method on a simple example 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.

6. Discussion

Our method serves as a tool for incorporating complex constraints in denoising diffusion processes, specifically when dealing with multi-constraint specifications with discrete solution spaces. Its iterative nature allows it to address nonlinear constraint problems and extends the power of denoising diffusion probabilistic models to work with constraints. There are various reasons our framework may help in modelling unknown discrete structures defined by constraint manifolds.

Sampling distinct solutions: When dealing with a set of constraints that has multiple distinct solutions, the discrete approximation M' provides a way to sample from these solutions. Each connected component in M' represents a distinct solution, and by sampling points within each component, we can obtain different feasible solutions to the given constraints. This is particularly valuable when exploring a problem space with diverse possible outcomes.

Discretization for efficiency: In many scenarios, working with continuous solution spaces can be computationally expensive or impractical. Discretizing the solution space using a finite set M' allows for more efficient computations. Sampling from the discrete approximation simplifies the problem by reducing it to a finite and manageable set of

points. This simplification can lead to faster computations and easier analysis of the solutions.

Identifying disconnected regions: The concept of topologically disconnected regions in M' helps identify distinct subsets of solutions that cannot be connected by continuous paths. By sampling from these disconnected regions separately, we can obtain solutions that are fundamentally different from each other. This is valuable when searching for diverse or non-redundant solutions.

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.

7. Appendix A: Constraint Induced Diffusion as Correlations

Consider, for instance, a scenario involving pairwise distance constraints between a set of variables denoted as $\boldsymbol{d} = d_{ij}$, where d_{ij} signifies the distance separating variables i and j. These constraints can be mathematically expressed through the set of functions $C_{ij}(\boldsymbol{\epsilon}) = ||\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}j|| - dij = 0$, which is applicable to all corresponding variable pairs $(i, j) \in \boldsymbol{d}$, influencing the samples drawn from a Multivariate Normal distribution.

The introduction of these geometric constraints essentially interrelates variables that were initially independent in the Gaussian distribution. In order to comprehend the implications of these constraints, the covariance matrix $\boldsymbol{\Sigma}'$ of the perturbed distribution $p'(\boldsymbol{\epsilon}')$ is worth examining:

$$\boldsymbol{\Sigma}' = \mathbb{E}_{\boldsymbol{\epsilon}' \sim p'}[\boldsymbol{\epsilon}'(\boldsymbol{\epsilon}')^T] - \mathbb{E}_{\boldsymbol{\epsilon}' \sim p'}[\boldsymbol{\epsilon}']\mathbb{E}_{\boldsymbol{\epsilon}' \sim p'}[\boldsymbol{\epsilon}']^T, \quad (9)$$

Here, the expectations are calculated over the perturbed distribution. The covariance matrix $\boldsymbol{\Sigma}'$ elucidates the correlations among variables that emerge as a result of the geometric constraints.

Importantly, these correlations, which are encoded within the covariance matrix of a multivariate Gaussian distribution, represent the constraints in the distribution. This provides a way to naturally incorporate constraint-based information into the model.

8. Appendix B: Generalized Schur Complement for Multiple Constraints

To obtain a generalized approach of Schur Complement for multiple distance constraints, let's consider a set of M pairwise constraints between atoms. We can express each constraint as a function of the positions of the corresponding atoms:

$$f_m(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||^2 - d_{ij}^2 = 0, \quad m = 1, 2, \ldots, M, \quad (10)$$

where d_{ij} is the distance constraint between atoms i and j.

To incorporate all the constraints, we can form the combined gradient and Hessian matrices by stacking the corresponding matrices for each constraint:

$$\nabla \mathbf{f} = \left[\nabla f_1 \, \nabla f_2 \vdots \nabla f_M\right], \tag{11}$$

$$\nabla^2 \mathbf{f} = \left[\nabla^2 f_1 \, \nabla^2 f_2 \vdots \nabla^2 f_M\right]. \tag{12}$$

To project the Gaussian distribution with the original covariance matrix $\mathbf{\Sigma}$ onto the space of distance constraints, we can use the following generalized Schur complement:

$$\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma}\nabla^2 \mathbf{f}^T (\nabla^2 \mathbf{f} \mathbf{\Sigma} \nabla^2 \mathbf{f}^T)^{-1} \nabla^2 \mathbf{f} \mathbf{\Sigma}. \tag{13}$$

While the Schur complement method can be implemented iteratively for non-linear systems, it is computationally intensive due to the inversion of the Hessian matrix. However, it serves as an excellent theoretical tool, providing a precise representation of how constraints can be formally incorporated into the diffusion process. On the other hand, the Schur complement method provides a direct way to project the covariance matrix of the atomic positions onto the space that satisfies the distance constraints. It essentially modifies the covariance matrix in a way that embeds the constraints, without needing to adjust the atomic positions. This approach formally modifies the probability distribution of interest, and may be more useful for theoretic insight.

9. Appendix C: 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 $\sigma_{jk} := \|x_i^l - x_j^l\| - d_{jk}$ which is $\leq or \geq 0$, one has

$$y = \begin{cases} max(0, ||x_i^l - x_j^l|| - d_{jk}^u), & \text{if } \leq \\ max(0, d_{jk}^l - ||x_i^l - x_j^l||), & \text{if } \geq \end{cases} \tag{14}$$

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.

10. Appendix D: Incorporation of Logical Operators in Geometric Constraints

The application of logical operators such as'AND', 'OR' and 'NOT' within geometric constraints enables a more flexible and representative modeling of physical and chemical systems. Real-world scenarios frequently require the satisfaction of multiple constraints following complex logical rules. Below, we detail the basic implementation of 'OR' and 'NOT' logical operators within the geometric constraints of our diffusion process while noting that the 'AND' operator is the basis of the formalism:

10.0.1. 'OR' Logic

The 'OR' condition necessitates that at least one of two (or more) constraints be met. Let's denote two constraint functions as $f_1(x)$ and $f_2(x)$. The 'OR' logic can be integrated by constructing a composite constraint function that is satisfied when any of its constituent constraints is met. We can express this as:

$$g(x) = \min(f_1(x), f_2(x)) \tag{15}$$

In this case, if either $f_1(x) = 0$ or $f_2(x) = 0$ (or both), $g(x) = 0$, thereby meeting the 'OR' condition. Alternatively, we can employ a product of the constraints:

$$g(x) = f_1(x) \cdot f_2(x) \tag{16}$$

If either $f_1(x) = 0$ or $f_2(x) = 0$ (or both), $g(x) = 0$, again adhering to the 'OR' logic. This method requires that both $f_1(x)$ and $f_2(x)$ are always non-negative.

10.0.2. 'NOT' Logic

The "NOT" operator in the context of geometric constraints could be defined using the following equations. Let's say we have a constraint $f(x) = 0$. We want to define a NOT operator for this constraint. We can then define "NOT f(x)" as regions where f(x) does not equal zero, which can be represented with two inequality constraints which can be combined via the 'OR' operator to designate the 'NOT' operator.

We denote ϵ as a small positive number, then "NOT f(x)" can be represented as:

$$g_1(x) = f(x) + \epsilon < 0 \tag{17}$$

$$g_2(x) = f(x) - \epsilon > 0 \tag{18}$$

In the equations above, we have defined two regions (when $f(x)$ is smaller than $-\epsilon$ and larger than ϵ) where "NOT

Algorithm 1 Pseudo-Code for Training

$t \sim U(0, T)$, $\epsilon \sim N(0, I)$
Subtract center of gravity from ϵ: $\hat{\epsilon} = [\epsilon(x), 0] - [x, 0]$
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 $\mathcal{L}_c = |\epsilon_s - \epsilon'_s|_2^2$

f(x)" is true, thus defining a NOT operator for our constraints. Note that these regions depend on the choice of ϵ.

11. Appendix E: Training and Sampling Algorithms

11.1. Training Process

During training, in Algorithm 1, 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.

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.

11.2. Generative Process

In this generative process, we 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.

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.