JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

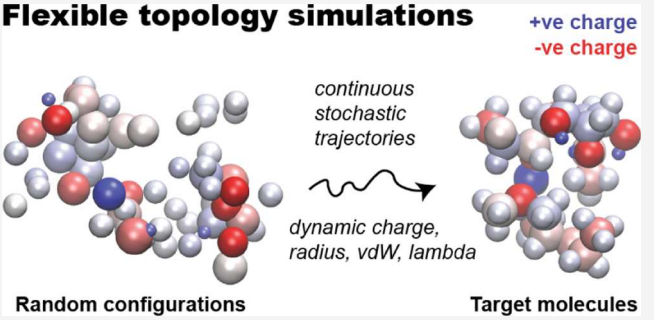
Flexible Topology: A Dynamic Model of a Continuous Chemical Space

Nazanin Donyapour, Fatemeh Fathi Niazi, Nicole M. Roussey, Samik Bose, and Alex Dickson*

Cite This: J. Chem. Theory Comput. 2023, 19, 5088−5098 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Ligand design problems involve searching chemical space for a molecule with a set of desired properties. As chemical space is discrete, this search must be conducted in a pointwise manner, separately investigating one molecule at a time, which can be inefficient. We propose a method called "Flexible Topology", where a ligand is composed of a set of shapeshifting "ghost" atoms, whose atomic identities and connectivity can dynamically change over the course of a simulation. Ghost atoms are guided toward their target positions using a translation-, rotation-, and index-invariant restraint potential. This is the first step toward a continuous model of chemical space, where a dynamic simulation can move from one molecule to another by following gradients of a potential energy function. This builds on a substantial history of alchemy in the field of molecular dynamics simulation, including the Lambda dynamics method developed by Brooks and co-workers [X. Kong and C.L. Brooks III, J. Chem. Phys. 105, 2414 (1996)], but takes it to an extreme by associating a set of four dynamical attributes with each shapeshifting ghost atom that control not only its presence but also its atomic identity. Here, we outline the theoretical details of this method, its implementation using the OpenMM simulation package, and some preliminary studies of ghost particle assembly simulations in vacuum. We examine a set of 10 small molecules, ranging in size from 6 to 50 atoms, and show that Flexible Topology is able to consistently assemble all of these molecules to high accuracy, beginning from randomly initialized positions and attributes.



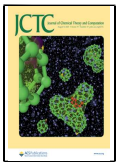
INTRODUCTION

Many problems in computational chemistry—such as searching for stable ligand-bound poses, or predicting the folded structure of a protein—involve searching "conformational space": a high-dimensional space where each point represents a molecular conformation. Conformational space could be searched manually, by making moves along individual degrees of freedom and assessing their impact, but this strategy would be ineffective for complex systems as biomolecular motions are often intricately coupled, combining both solute and solvent degrees of freedom. Molecular dynamics (MD) is a useful tool to navigate conformational space; by following the high-dimensional gradients of the energy function, it naturally generates and harnesses these collective motions. Although trajectories can sometimes get caught in local energy minima, there have been a myriad of advancements in algorithms that can efficiently use dynamics to sample even very rough energy landscapes.[1−12]

A similar concept invoked in the field of drug discovery is "chemical space": a space that contains the set of all possible chemical structures that could in theory be developed as small-molecule ligands to bind a given target.[13] This can be used to visualize different steps of drug discovery, where, say, an initial virtual screen evaluates activity at a large number of points scattered around the space. Hit-to-lead (H2L) and lead optimization typically involve a series of small chemical changes to a molecule with known activity, which can be thought of as steps along a path that winds through chemical space. Of course, chemical space and conformational space are coupled, as chemical changes to the ligand can induce structural changes in a protein binding site.[14−16] Conversely, protein conformational changes—for example, those induced by complexation with cofactors, or post-translational modifications—can also affect the relative activity of different ligands.[17,18]

In contrast to conformational space, chemical space is discrete and cannot be explored in a continuous fashion. This typically limits chemical space exploration to pointwise investigations in which the conformational space is explored separately for each ligand. A notable exception to this is alchemical methods,[19] where an external parameter, normally denoted by λ, is used to "turn on" or "turn off" additional terms

Received: April 11, 2023
Published: July 24, 2023



in a potential energy function: $U(X, \lambda) = U_0(X) + \lambda U_{pert}(X)$. Alchemical approaches have been used to calculate free energy differences between protonated and deprotonated states[20−22] and are commonly used to calculate both relative and absolute free energies of binding.[23−25] These methods form continuous paths in chemical space that connect either two compounds or connect one compound with the "null" state, a point in chemical space representing the absence of a ligand. This approach has also been used in the lambda dynamics method,[26−31] where each λ is considered as a dynamical variable. That is, λ feels a force according to: $F_\lambda = -\partial U/\partial\lambda$, and has a velocity v_λ and a mass m_λ. In chemical space, the lambda dynamics method provides a network of continuous paths between structures using a large number of λ variables that control the presence of different ligand side chains.[32−34] However, as λ-based methods are limited to interpolating between structures, we set out to create a more general model of a continuous chemical space that could be used for chemical discovery.

To enable such an algorithm, we need to introduce the concept of flexible molecular topology. In the framework of a classical MD simulation, "topology" describes the identities of each atom and their covalent connectivity in the system. The atomic identities are extensive; for instance, the CHARMM36 protein force field has 20 different types of carbon where each atom type has unique parameters for the partial charge, bond lengths, and rotation around chemical bonds.[35] A harmonic energy function defines the bonds between atoms, which can vibrate around a minimum energy position but can never break. The system's topology for the classical MD simulation is thus determined at the beginning of the simulation and remains fixed for the duration of a trajectory.

Here we introduce the Flexible Topology method, which allows the topology to change over the course of the simulation by treating the attributes that define atomic identities as dynamic variables. To enable the continuous and dynamic description of chemical identities in the Flexible Topology method, we separate external interactions from internal ones and treat them separately. While internal interactions are complex and involve a multitude of atom-type-specific parameters, external interactions are composed of only Lennard-Jones and electrostatic interactions that can be wholly described by only three parameters: partial charges (q), Lennard-Jones well depths (ϵ) and Lennard-Jones radii (σ). In Flexible Topology, each atom is assigned a dynamic λ variable, as well as three parameters q_i, ϵ_i, and σ_i. These are all treated as dynamical variables that change as a function of time and together we refer to them as "attributes" denoted by $A_i = \{q_i, \epsilon_i, \sigma_i, \lambda_i\}$ for atom i and A_{GP} for the complete set. These variables, like λ, have masses (m_q, m_ϵ, and m_σ) and their motion follows the gradients of an extended Hamiltonian energy function. These dynamic variables are used to model nonbonded interactions of the shapeshifting ligand atoms, which we refer to as "ghost particles", using conventional Lennard-Jones and Coulomb equations. The ghost particle positions and attributes both change together over the course of a trajectory and are influenced both by their surrounding environment and symmetry-invariant restraint terms that encourage the particles to assemble to one or more target molecular structures. Importantly, the environment—binding site residues and water molecules—can in turn respond to the presence of the ghost particles, providing a mechanism to model ligand-induced conformational change.

In this paper, we introduce the Flexible Topology method and assess its performance on small ligand-only systems, determining suitable algorithmic parameters to help pave the way for more complex applications of the method. We first describe the mathematical framework of Flexible Topology and explain its implementation in the OpenMM molecular dynamics package. We define atomic environment vectors (AEVs) using functions from Behler and Parrinello that are used to both (1) map (or "assign") ghost particles to atoms of the target ligand and (2) define the loss function that guides the ghost particles toward the target structures. We find that for small ligands (20 atoms or less), this loss function is sufficient to reliably guide the assembly of the target ligand. For larger ligands, additional "second-order" restraints that bring together specific pairs of atoms are useful to guide assembly. To thoroughly assess the performance of the algorithm and determine suitable parameters, we run assembly simulations with a set of 10 target ligands in vacuum and define an appropriate metric for the quality of assembly. Finally, we conclude with a discussion of the challenges and future work of the Flexible Topology method, looking toward applications such as the prediction of bound poses, calculation of relative binding free energies, and simultaneous exploration of chemical and conformational space.

METHODS

Framework of Flexible Topology. The goal of the Flexible Topology method is to model a set of atoms that can continuously transform between different molecules. To implement this method, we use MD simulations that are coupled with an ML-based external force (MLForce)[36] that slowly nudges a set of ghost particles to become a drug-like molecule (Figure 1).

MLForce consists of three components: a molecular representation model, an assignment algorithm, and a loss function. The molecular representation model defines a set of symmetry-invariant atomic environment vectors (AEVs) using functions introduced by Behler and Parrinello.[37] The AEVs,

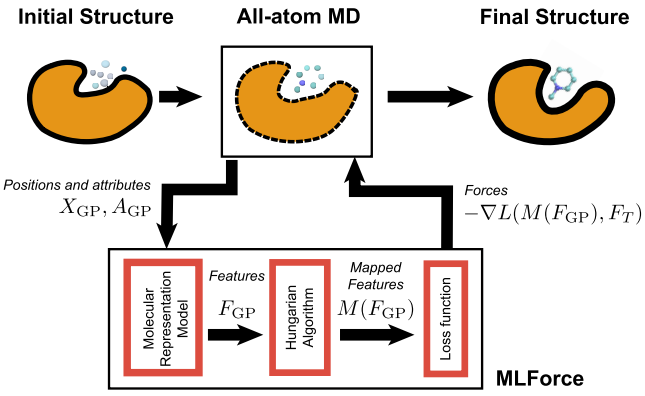


Figure 1. Framework of the Flexible Topology method. The gray and blue spheres show the ghost particles. MD simulations start with an initial structure of the protein of interest and randomly initialized ghost particles. The MLForce plugin computes external forces on the ghost particle positions and attributes using a loss function, L. X_{GP} denotes the ghost particle positions, A_{GP} represents the atomic attributes of ghost particles, M shows the mapping function, and F_{GP} and F_T show ghost particle and target ligand features, respectively. Final output shows the ghost particles assembled into a target structure in the binding pocket of the protein.

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

concatenated with the ghost particle attributes, determine the set of features for each atom that describe both its environment and its identity. We discuss these aspects in detail in the next section.

The second component is an algorithm to solve an assignment problem to determine an optimal mapping between the ghost particles and target ligand atoms. In our implementation, we employ the Hungarian algorithm[38] to find an assignment with a minimum cost between the ghost particles and target ligand atoms. The cost matrix is determined by calculating the Euclidean norm between the ghost particle features (F_{GP}) and target molecule features (F_T). The Hungarian algorithm has polynomial time complexity and we used a C++ implementation from Cong Ma (The original code is implemented in Matlab by Buehren) https://github.com/mcximing/hungarian-algorithm-cpp.git. The output of this algorithm is a mapping M that minimizes the cost. The cost, which we refer to as the "loss function" is defined as

$$L = \sum_{i=1}^{N} |F_{GP,M(i)} - F_{T,i}|^2 \equiv |M(F_{GP}) - F_T| \tag{1}$$

where || on the right denotes the squared L2 norm computed over all of the atoms.

The loss function L is used to define the restraint potential that guides the ghost particles to adopt their target configurations. Gradients of L yield external forces on both the positions and attributes of the ghost particles. These forces are applied to the ghost particles to change their positions and attributes at each dynamic step, gradually transforming them into a drug-like molecule in an orientation that fits the protein of interest. This loss function is discussed in detail in the Loss Function and Flexible Topology Restraint Energy section.

Implementation of the MLForce plugin follows the architecture of the OpenMM library.[39] It includes multiple layers, which are as follows: (1) an OpenMM public Application Programming Interface (API) to generate the MLForce object with an embedded PyTorch model, (2) a platform-independent layer to load the model, and (3) a computational kernel layer for executing the model and computing the gradients. All of the layers are written in C++ and the computational kernels are implemented for CUDA, OpenCL, and Reference (CPU) platforms in OpenMM. Here, we utilize TorchANI[40] a PyTorch[41] implementation of the Behler−Parrinello symmetry functions to compute features, and we use Torch functions to calculate the loss function derivatives with respect to the positions of ghost particles.

Symmetry-Invariant Molecular Features. The atomic environment vectors that guide the assembly of the ghost particles are mostly composed of a customized set of Behler-Parinello symmetry functions.[37] These functions capture the radial and angular information of the surrounding atoms for each individual atom. Each feature is invariant to translation and rotation operations and are differentiable with respect to atomic positions. The local radial environment of an atom is defined by the sum of Gaussian terms multiplied by a cutoff function

$$G_i^R = \sum_{i \neq j} e^{-\eta_R (R_{ij} - R_s^R)^2} f_c(R_{ij}) \tag{2}$$

where R_{ij} is the distance between atoms i and j, and f_c is the cutoff function that goes from 1 to 0 as R_{ij} surpasses a cutoff distance R_c to approximate local atomic environment. The parameter η_R controls the width, and R_s^R determines the location of the Gaussian functions. Using a set of different values for R_s^R allows for capturing information in particular regions of the radial environment around i. The cutoff function is defined as follows

$$f_c(R_{ij}) = \begin{cases} 0.5\left(\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1\right) & \text{for } R_{ij} \leq R_c \\ 0.0 & \text{for } R_{ij} > R_c \end{cases} \tag{3}$$

where R_c is the cutoff distance.

The angular functions are defined for an atom i using the sum of cosine terms for all possible angles centered at atom i

$$G_i^A = 2^{1-\zeta} \sum_{j,k \neq i} (1 + \cos(\theta_{ijk} - \theta_s))^\zeta e^{-\eta_A \left(\frac{1}{2}(R_{ij} + R_{ik}) - R_s^A\right)^2} f_c(R_{ij}) f_c(R_{ik}) \tag{4}$$

where η_A and R_s^A have the same role to modify the width and location of radial Gaussian functions, and the parameters ζ and θ_s control the width and location of different angular regions. Note that these angular functions are the modified versions of original Behler-Parinello symmetry functions[37] and reflect the changes made by Smith et al.[42] Using multiple values of η_R, R_s^R, η_A, R_s^A, θ_s, and ζ enables us to construct a set of symmetry functions that can describe the local environment of an atom in high detail.

Although we employ the TorchANI module[40] to compute the symmetry functions, there are some significant differences in our implementation. First, whereas in ANI, atoms have a fixed elemental type (e.g., H, C, N, or O), here we consider all of the atoms to have the same type. However, we found that symmetry functions with one atom type struggle to couple chemical and geometrical information properly in some cases. Therefore, we added an additional set of radial functions to describe the atomic environment not only in terms of density but also atomic charge. We employ the radial function defined in eq 2 while using the following cutoff function that considers the charge of the pair atom (q_j)

$$f_{cq}(R_{ij}) = \begin{cases} 0.5\left(\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1\right) q_j & \text{for } R_{ij} \leq R_c \\ 0.0 & \text{for } R_{ij} > R_c \end{cases} \tag{5}$$

The resulting features are denoted G_i^{Rq}.

The full set of features for atom i is then: $F_i = (\{G_i^R\}, \{G_i^{Rq}\}, \{G_i^A\}, \{A_i\})$, where the $\{\}$ brackets for the G functions denote a list over all permutations of the parameters ζ_R, R_s^R, η_A, R_s^A, θ_s, and ζ. $\{A_i\}$ is the set of the four attributes for atom i, namely, q_i, ϵ_i, σ_i, and λ_i. The parameter values used to construct the features are given in Table 1. In total, there are 24 radial features and 128 angular features, along with another 24 radial features using the charged cutoff function f_{cq}, and the four atomic attributes, for a total of 180 features per atom.

We denote the list of the ghost particle feature vectors as F_{GP}. Note that while F_{GP} is invariant to translation and rotation, it is not invariant to the re-indexing of atoms. However, the mapped set of features, $M(F_{GP})$ is symmetric to re-indexing. In this way, our loss function, energies, and forces are invariant to all three symmetries.

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

Table 1. Parameter Values for Behler−Parrinello Functions, Grouped into Radial and Angular Components[a]

	var.	values	n_{vals}	N_{feat}
radial	η_R (nm^{-2})	1600	1	24
	R_s^R (nm)	0.090, 0.169, ⋯, 0.708	24	
	R_c^R (nm)	1.0	1	
angular	η_A (nm^{-2})	800	1	128
	R_s^A (nm)	0.090, 0.155, ⋯, 0.545	8	
	θ_s^A (radians)	$\pi/16$, $2\pi/16$, ⋯, π	16	
	ζ	32	1	
	R_c^A (nm)	0.5	1	

[a]The number of values used for each parameter is shown in the n_{vals} column. The product of n_{vals} is the total number of features for each component (N_{feat}).

Loss Function and Flexible Topology Restraint Energy. Note that although we refer to the value in eq 1 as a loss function, and it is computed from a PyTorch model, it currently does not have any trainable parameters and thus requires no training data. The target features F_T are precomputed prior to the simulation and the ghost particle features F_{GP} are deterministic functions of the coordinates X_{GP}. This loss is transformed into a perturbative energy according to

$$U_{FT}(X_{GP}, A_{GP}) = \frac{s}{\beta} L(X_{GP}, A_{GP}) \tag{6}$$

where we explicitly show the loss and potential energy as a function of the ghost particle coordinates (X_{GP}) and attributes (A_{GP}). β is the inverse temperature, and s is a unitless parameter called the "MLForce Scale" that scales the Flexible Topology energy against the other energies in the system. We then add this term to the system energy function U

$$U(X, X_{GP}, A_{GP}) = U_0(X) + U_{nb}(X, X_{GP}, A_{GP}) + U_{FT}(X_{GP}, A_{GP}) \tag{7}$$

where $U_{nb}(X, X_{GP}, A_{GP})$ is the nonbonded interaction energy between the ghost particle atoms and the remainder of the system (X). The forces on the positions and attributes of the ghost particles are determined by the negative gradients of U. Eq 7 makes clear that these depend both on U_{FT} (and in turn, the loss function L) and U_{nb}. The relative contributions of these forces to the motion of the particles and attributes are again determined by the parameter s.

Simulation Details. The simulations in this work are conducted in vacuum, where the Flexible Topology particles are the only particles in the system. To initialize the system, we begin by randomly introducing 22 particles in a 6 by 6 by 6 Å cube. Ghost particles are inserted one at a time and trial positions are only accepted if the atom is farther than 0.4 from all system atoms and all ghost particles inserted so far. Ghost particle attributes σ, ϵ, and q are selected randomly from the bounds in Table 2. The λ attributes are initialized uniformly at 0.7.

The simulations are conducted using OpenMM.[39] To create a simulation object, first empty system and topology objects are created and the ghost particles are added. To ensure adequate stability at the desired simulation conditions (a timestep of 2 fs and a temperature of 300 K), each particle is given a mass of 100 amu. A number of forces are then added to the system as described below.

Table 2. Minimum and Maximum Parameter Values for Ghost Particle Attributes

	min.	max.
q (Debye)	−0.7	0.7
σ (nm)	0.022	0.23
ϵ (kJ/mol)	0.037	2.63
λ	0.0	1.0

A custom external force that we call a "continuity force" is added, which ensures that the particles form an unbroken network of interactions. More precisely, if we imagine the particles as nodes of a network and draw an edge between particles that are closer than a cutoff distance (d), then the nodes would all be in the same component of the network. In other words, there exists a path of some length between any two nodes i and j. The continuity force achieves this by classifying particles into components on the fly and adding an attractive force between pairs of particles belonging to different components. The particles are chosen for each dynamics step as the closest possible pairs between the two components. The attractive force is of the form $E(r) = k(r - d)^2$, where r is the separation between the particles and d is a user-defined cutoff distance. Here, we employ d = 0.18 nm and k = 100 kJ/mol/nm^2, which was sufficient to ensure that the particles do not fly away from each other, even when the MLForce is inactive (s = 0).

The custom force called MLForce that implements the loss function was then added to the system. This is initialized using a PyTorch model to calculate the AEVs, loss function values, and derivatives, as well as a set of target features. This can also include a set of distance restraints between ghost particles that we refer to as "second-order restraints" since they use assignments of two atoms to define a restraint energy. Unless otherwise noted below, we use a distance restraint between every pair of atoms in the molecule that takes a harmonic form: $E(r) = k_r(r - r_0)^2$, where r_0 is the distance between those atoms in the target structure and k_r is a force constant. Note that the s parameter (eq 6) modulates the entire MLForce energy, including the second-order restraints.

Lastly, we add a separate nonbonded force for the ghost particles to ensure that they do not overlap in space. This is a purely repulsive version of a Lennard-Jones force: $E_{LJ-rep} = \epsilon (r/\sigma)^{12}$, where ϵ and σ are not related to the dynamic particle attributes and are simply constant values of 2.0 kJ/mol and 0.4 Å, respectively.

Note that if there were other nonghost particles in the system (e.g., water molecules, proteins, or other solutes), then additional nonbonded forces would be required that take into account the dynamical attributes ($\{\sigma_i\}$, $\{\epsilon_i\}$, $\{q_i\}$). While these are not included in the work presented here, examples that implement these interactions are the subject of future work and are already available to the reader in the Flexible Topology repository.[43]

The dynamics of both the positions and the attributes are propagated by a custom OpenMM integrator. The source code for the integrator (`CustomHybridIntegratorConstCharge`) is given in the Flexible Topology code repository (`src/flexibletopology/utils/integrators.py`). For this integrator, Langevin dynamics with a friction coefficient of 1 ps^{-1} and an integration step size of 2 fs is used for advancing the atomic positions. Each atomic attribute (a) is evolved using Brownian dynamics as follows

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

$$a_{t+\delta t} = a_t + \delta t\left(-\frac{\partial U}{\partial a}\frac{1}{\gamma m} + \sqrt{\frac{2kT}{\delta t \gamma m}}R_t\right) \quad (8)$$

where δt is the timestep and $R(t)$ is a memory-less Gaussian process with mean 0 and variance 1. Note that for each attribute only the product of the friction coefficient (γ) and attribute mass (m) is required. Appropriate values of these parameters were determined by equalizing the relaxation timescales of the attributes and are reported in Table 3. As each attribute has different units, the units of γm differ for each attribute and are also given in Table 3.

Table 3. Parameters Governing the Forces and Dynamics of Particle Attributes[a]

	γm	units	Φ	units
q	10000	$\left[\frac{\text{kJ}}{\text{mol}}\right]$[ps][Debye^{-2}]	40000	[Debye^{-2}]
σ	10000	$\left[\frac{\text{kJ}}{\text{mol}}\right]$[ps][nm^{-2}]	50000	[nm^{-2}]
ϵ	10000	$\left[\frac{\text{kJ}}{\text{mol}}\right]^{-1}$[ps]	1000	$\left[\frac{\text{kJ}}{\text{mol}}\right]^{-2}$
λ	10000	$\left[\frac{\text{kJ}}{\text{mol}}\right]$[ps]	20000	[]

[a]Friction coefficient times mass (γm).

The term $\frac{\partial U}{\partial a}$ can be expressed as the sum of two terms

$$\frac{\partial U}{\partial a} = \frac{\partial U_{\text{nb}}}{\partial a} + \frac{\partial U_{\text{FT}}}{\partial a} \quad (9)$$

To balance these with respect to one another, an additional scaling factor (Φ_a) is used within the loss function

$$L_A = \sum_i \sum_{a\in A} \Phi_a (f_{\text{GP},a,M(i)} - f_{\text{T},a,i})^2 \quad (10)$$

where L_A represents the component of the loss function arising from discrepancies in the attributes, A is the set of attributes (q, σ, ϵ, λ), $f_{\text{GP},a,M(i)}$ is the value of attribute a of the ghost particle that maps to target atom i, and $f_{\text{T},a,i}$ is the value of attribute a of target atom i. Appropriate values for Φ_a were determined by increasing their value until adequate convergence of the attributes was achieved. The factors $\{\Phi_a\}$ are given in Table 3. Finally, another feature of the integrator used here is that it keeps the sum of the ghost particle partial charges equal to a constant value, which is set to zero for all molecules simulated here, but could be set to an integer value for charged molecules.

RESULTS

Assembly Simulations. As it is guaranteed that $M(F_{\text{GP}}) = F_{\text{T}}$ marks the global minimum of U_{GP} and that the influence of U_{GP} can be made arbitrarily strong by the s parameter, successful assembly is not guaranteed. First, it is possible that the set of molecular features is not descriptive enough to reproduce the target structures to high enough accuracy, that is, the system could obtain very low loss values while still not reaching low root-mean-square distances (RMSD) to the target structure. Second, it is possible that U_{GP} is not smooth with respect to X_{GP} and A_{GP}, and that the particles get caught in local energy minima. To address both of these possibilities, we run assembly simulations in vacuum where the positions and attributes of the ghost particles are randomly initialized. In this approach, we employ the loss function in eq 1, with a single set of target features. We initialize s ("MLForce Scale") at zero and progressively increase it following a sigmoidal function to a maximum value s_{max}. Ideally, as the U_{GP} term grows stronger, the attributes and orientation of the ghost particles will gradually transform to resemble those of the target molecule. This is quantified by the loss per particle (L/n) and by the assembly root-mean-squared deviation (RMSD), which employs the atom assignments to assess the global structure of the molecule. One difficulty with the assembly RMSD is to take into account atoms that have identical or almost-identical atomic environment vectors, arising from symmetry or pseudo-symmetry in the target molecule. To account for this we enumerate all sets of atoms with near-identical AEVs, using a cutoff of $\sum_i (f_i - f_j)^2 < 3.0$, and create an exhaustive list of mappings using each possible permutation of equivalent atoms. We then compute the RMSD using each mapping. As all of the features are nonchiral—that is, they are invariant to reflection—we also compute the RMSD to a reflected copy of the target molecule for each mapping. The lowest RMSD value is then used as the assembly RMSD.

We randomly selected 10 molecules (Figure 2) from the OpenChem dataset[44] to use as target structures in the assembly simulations. These molecules range from 6 to 50 atoms in size and vary widely in their structures. The set includes heterocycles, branched hydrocarbons, and many examples of both symmetry and pseudo-symmetry, thus providing a challenging test set for assembly. Examples of time series analysis of the assembly simulations for three of these molecules are given in Figure 3. These simulations were conducted with a maximum MLForce Scale of 5000, which was increased over 100 cycles (of 500 dynamics steps each) and then held constant for an additional 50 cycles. A "two-body" distance restraint was used with a force constant of 0.5 kJ/mol/nm^2; note that the effective strength of this term is also modulated by the MLForce Scale. Lastly, for these results, the assignments were determined once every 10 cycles, or once every 10 ps.

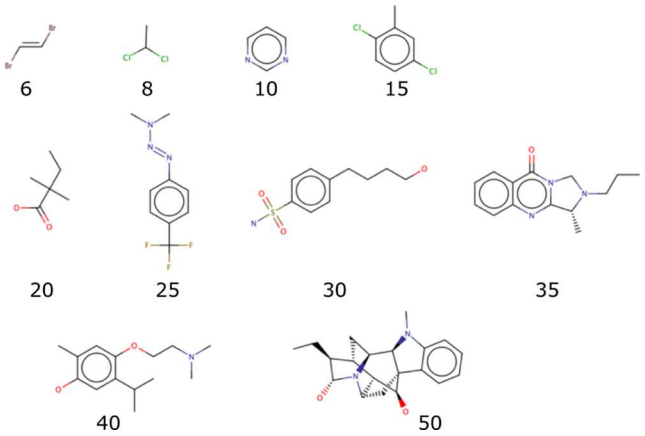


Figure 2. Assembly simulation target molecules randomly chosen from the OpenChem dataset[44] are shown in their 2D structures. The number of atoms for each molecule is written under it. The 2D structures are generated and drawn by MarvinSketch using implicit hydrogens.

In all three of the time series plots in Figure 3, the loss per particle roughly corresponds to the assembly RMSD and both quantities decrease over the course of the simulation in a fashion that roughly mirrors the sigmoidal increase of the

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

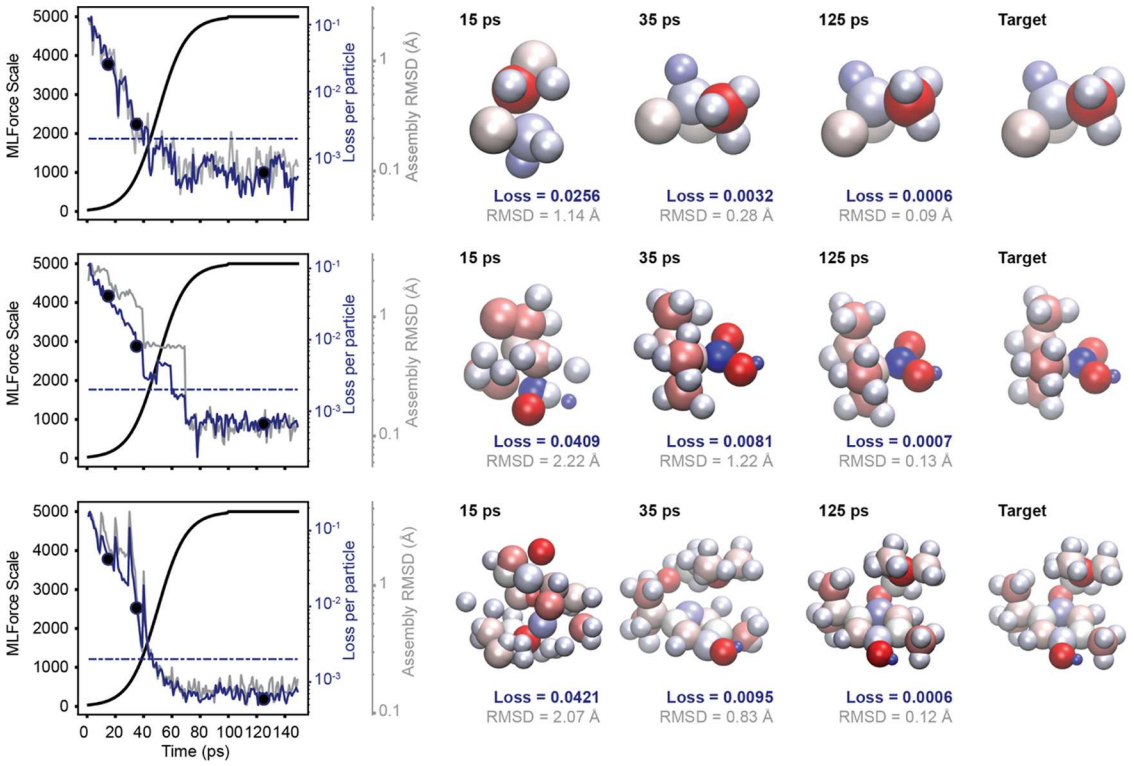


Figure 3. Time series analysis of the per particle loss (blue) and the assembly RMSD (gray) for successful assembly trajectories for molecule 8 (top), 20 (middle), and 40 (bottom). 2D structures are shown in Figure 2. A dashed line showing a loss-per-particle value of 0.002 is shown on each graph, which can be used across systems as an indicator of excellent assembly accuracy. The structures on the right show snapshots of the molecule at intermediate stages of assembly and correspond to the three black circles on the graphs on the left-hand side. The structures are shown in size-charge representation, where the radius of each particle roughly corresponds to its σ value and the color shows its charge (q), with blue being positive, red being negative, and white being neutral. The target structures show the conformers that were used to generate the target set of features that are almost indistinguishable from the 125 ps structures in each case.

MLForce Scale when viewed on the logarithmic scale. Significant dips or spikes in the loss per particle are more often than not reflected in the assembly RMSD as well, although in some cases decreases in the loss per particle are not reflected in the assembly RMSD, which could possibly be the result of inaccuracies in atom assignments between pseudo-symmetric atoms with $\sum_i (f_i - f_j)^2 > 3.0$ and thus not considered by our RMSD calculations. For this reason, we use the loss per particle as our standard measure of accuracy. In these simulations, we have found that a useful threshold for separating "good" from "bad" assembled structures is a loss per particle value of 0.002, which is marked on the time series plots in Figure 3 by a horizontal dashed line. Across all of the examples considered here, we find that 89% of the snapshots with $L/n < 0.002$ also have very high structural fidelity (RMSD < 0.5 Å) (Figure S1), addressing the first concern outlined at the beginning of this section.

To more rigorously assess the assembly performance across a range of different parameters, we selected a central point in "parameter space" and examined a series of perturbations along individual parameters. For each set of parameters, a set of 10 assembly simulations was performed for each of the targets in Figure 2. The quality of the assembly simulations was assessed using the minimum value of the loss per particle obtained during the assembly trajectory. Figure 4 shows a series of box plots of the minimum loss per particle. The horizontal black line marks our quality threshold of 0.002, and in general, we observe that assembly for smaller targets is more robust than assembly of larger targets, as expected.

Figure 4A examines the number of steps per cycle. A larger number of steps per cycle gives the particles more time to adapt to the gradually increasing restraint forces that encourage assembly; however, they also increase the computational cost of the simulation. As all simulations here use 2 fs timesteps and 150 cycles, the total length of the trajectories ranges from 30 ps (for 100 steps/cycle) to 150 ps (for 500 steps/cycle). Generally, we observe an increase in the quality of the assembly with increasing steps per cycle, with 500 steps (green bars) showing median values that are at or below our quality threshold for all of the cases examined here.

As discussed above, the assignment algorithm uses the current values of the atomic environment vectors to match the ghost particles to atoms in the target molecule. Figure 4B examines the periodicity of the assignment calculation, including both extreme cases: fixing the assignments throughout the entire simulation ("fixed", pink) and allowing the assignments to change at every dynamics step ("1", yellow). We find, surprisingly, that fixing the assignments gives adequate performance, and too-frequent assignments can lead to poor assembly performance, especially in larger molecules. Computing assignments every 5000 dynamics steps (blue) also yields good accuracy, while computing assignments every 500 steps (green) did not result in the successful assembly of the larger targets.

All assignments used a sigmoidal MLForce Scale scheme, as shown in Figure 3. To examine the impact of MLForce Scale on assembly accuracy, we conducted a set of simulations where the maximum value of this sigmoid was scaled to either 500,

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

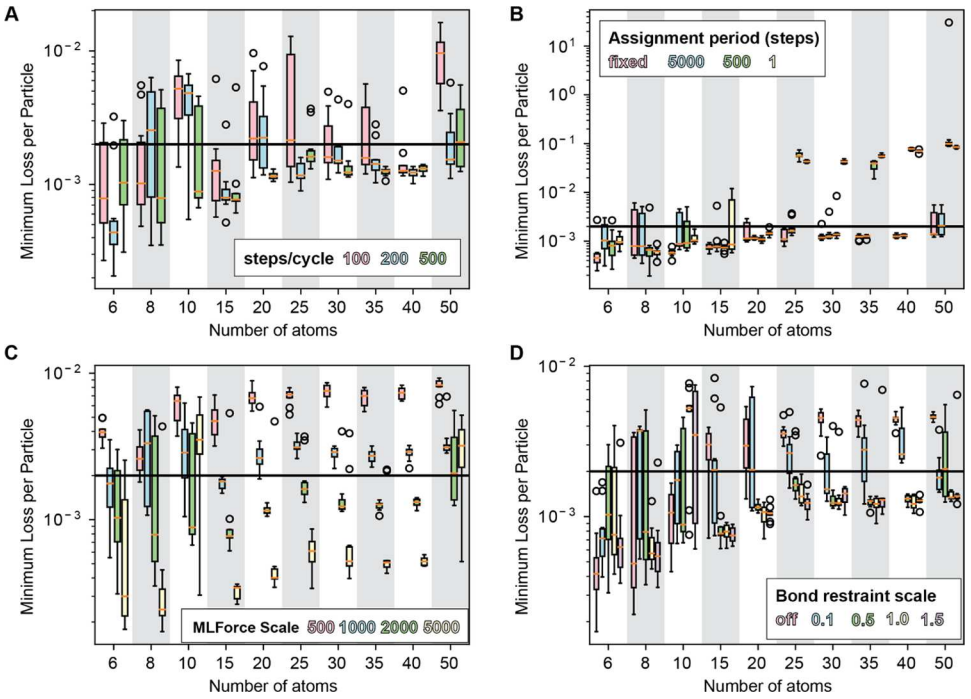


Figure 4. Performance of the Flexible Topology algorithm upon parameter perturbation. Each panel shows a set of box plots that show the minimum loss per particle achieved in a set of 10 independent assembly trajectories for each target. The boxes show the interquartile range, whiskers show 150% of the IQR, and outliers are shown as circles. The orange lines in each box mark the median datapoint of each set. A single parameter is explored in each panel: the number of steps per cycle (A), the periodicity of atom assignment (B), the value of the MLForce Scale (C), and the strength of the second-order bond restraints (D). All parameter values were perturbations from the central point of: 500 steps per cycle, 5000 step assignment period, 2000 MLForce Scale, and 0.5 bond restraint strength.

Table 4. Flexible Topology Assembly Performance for Different Parameter Sets[a]

				small targets (≤20 atoms)				large targets (>20 atoms)			
C_{max}	k_r	n_{steps}	T_{update}	$\langle L_{min}/n \rangle$	$\langle L_{avg}/n \rangle$	$\langle R_{min} \rangle$	$\langle R_{avg} \rangle$	$\langle L_{min}/n \rangle$	$\langle L_{avg}/n \rangle$	$\langle R_{min} \rangle$	$\langle R_{avg} \rangle$
2000	**0.5**	**500**	**5000**	**0.0016**	**0.0030**	**0.19**	**0.43**	**0.0018**	**0.0030**	**0.29**	**0.51**
2000	0.5	100	1000	0.0026	0.0047	0.54	0.73	0.0041	0.0094	0.59	0.88
2000	0.5	200	2000	0.0024	0.0047	0.56	0.76	0.0016	0.0027	0.41	0.55
2000	**0.5**	**500**	**fixed**	**0.0012**	**0.0019**	**0.18**	**0.28**	**0.0015**	**0.0020**	**0.22**	**0.28**
2000	0.5	500	500	0.0011	0.0027	0.15	0.36	0.6610		2.14	
2000	0.5	500	1	0.0016	0.0053	0.18	0.55	0.0592		2.94	
500	0.5	500	5000	0.0049	0.0108	0.36	0.92	0.0073	0.0022	0.62	1.43
1000	0.5	500	5000	0.0026	0.0052	0.26	0.62	0.0030	0.0075	0.38	0.75
5000	**0.5**	**500**	**5000**	**0.0013**	**0.0020**	**0.20**	**0.33**	**0.0012**	**0.0016**	**0.30**	**0.41**
2000	off	500	5000	0.0018	0.0027	0.76	1.03	0.0043	0.0050	3.20	3.70
2000	0.1	500	5000	0.0023	0.0032	0.52	0.83	0.0026	0.0043	0.71	1.15
2000	**1.0**	**500**	**5000**	**0.0017**	**0.0037**	**0.14**	**0.32**	**0.0013**	**0.0026**	**0.22**	**0.39**
2000	1.5	500	5000	0.0014	0.0043	0.13	0.32	0.0015		0.24	
5000	0.5	100	1000	0.0028	0.0041	0.59	0.71				
5000	0.5	200	2000	0.0023	0.0036	0.55	0.65	0.0012	0.0018	0.46	0.53
5000	0.5	100	5000	0.0027	0.0030	0.37	0.50	0.0019	0.0022	0.43	0.51
5000	**0.5**	**200**	**5000**	**0.0019**	**0.0024**	**0.27**	**0.39**	**0.0016**	**0.0022**	**0.40**	**0.53**
5000	**0.5**	**1000**	**5000**	**0.0010**	**0.0019**	**0.16**	**0.28**	**0.0009**	**0.0013**	**0.25**	**0.36**

[a]Performance is measured separately on the small (≤20 atoms) and large (>20 atoms) targets, which contain five compounds each. The ⟨⟩ notation denotes averages over the five compounds and the ten trajectories per compound. L_{min} and R_{min} are the minimum values of the loss and RMSD obtained in a trajectory, respectively. L_{avg} and R_{avg} are the average values of those quantities obtained in the final 50 cycles of a trajectory, respectively. Parameter sets above the horizontal line correspond to the data shown in Figure 4. Dashes indicate that instabilities in one or more simulations caused exploded conformations that distorted the data.

1000, 2000, or 5000 (Figure 4C). Here, we find a very clear trend of decreasing per particle loss values with increasing MLForce Scale. While in some cases a maximum value of 2000 for the MLForce Scale (green) appears to be sufficient, a value of 5000 (yellow) consistently leads to better performance, which is particularly clear when examining the average value of the loss per particle over the final 50 cycles of the trajectory (Figure S2).

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

Finally, Figure 4D examines the "second-order" bond restraints that act on the distances between pairs of assigned atoms. Turning these restraints off has little or no effect on the assembly of smaller targets, but we find that a restraint strength of at least 0.5 is needed to reliably assemble targets of 20 atoms or more. However, we also find that these restraints can hinder the assembly of some smaller targets, particularly molecule 10, which has high symmetry. We also have observed some numerical instabilities in simulations with bond restraint values > 1.0. For this reason, we see a value of 0.5 to be a good compromise.

The performance of all of these parameter sets is summarized in Table 4. A number of parameter sets show excellent performance ($\langle L_{min}/n \rangle < 0.002$) on both large and small targets, which are highlighted in bold. However, L_{min}/n only requires a single trajectory frame to achieve a low loss value. All parameter sets that satisfy the more stringent criteria ($\langle L_{avg}/n \rangle < 0.002$) were obtained with a maximum MLForce Scale value of 5000. To further probe the performance of Flexible Topology at this parameter value we ran an additional series of simulations (below the horizontal line in Table 4). We find that the best performance is obtained by increasing the number of steps per cycle, but that excellent performance can still be obtained with as little as 200 steps per cycle, or 60 ps for each assembly trajectory.

For some combinations of parameters, we found that the trajectories were unstable: integration errors due to the finite timestep would accrue and compound catastrophically, as is common in molecular dynamics simulations that are performed with a step size that is too high for a particular energy landscape and temperature. The two biggest drivers of instability were the frequency of assignment updates and the strength of the second-order distance restraints. This is intuitive, as the assignment algorithm used here only takes into account the similarity of the atomic environment vectors between the ghost particles and the target atoms. Changes in the assignments could thus introduce discontinuities in the energy and energy gradients that are destabilizing. To minimize the impact of these assignment changes, we recommend that assignment updates are carried out relatively infrequently; we found 5000 steps to be a suitable interval for assignment updates.

DISCUSSION AND CONCLUSIONS

Here we have presented a novel method for molecular dynamics simulation called Flexible Topology that allows for both atomic connectivity and atomic identity to change over the course of a molecular simulation. In Flexible Topology, the identity of atoms is defined by a set of four continuous dynamical variables: partial charge (q), particle size (σ), and Lennard-Jones well depth (ϵ), as well as presence variable (λ) for each atom. These are guided to assemble into a target ligand using a deterministic loss function defined by the difference of atomic environment vectors (AEVs) of the atoms with the AEVs of the target atoms. The AEVs are modified versions of Behler−Parrinello symmetry functions,[37] which describe the local environment around each atom in a way that is invariant to translation and rotation operations. Derivatives of the loss function are used to define atomic forces that are added to an OpenMM MD simulation, implemented using a PyTorch model that incorporates features from the TorchANI[40] package. This was achieved with an external force plugin for the OpenMM tool, referred to as "MLForce",[36] which has already been made freely available on Github. MLForce is responsible for loading the TorchScript file of a TorchANI model and computing the internal force on the ghost particles. This approach is agnostic to the content of the TorchScript file; it can be easily swapped out with a model that computes different AEVs, or with a model that implements a different loss function that involves a deep neural network.

One of the main challenges of this project was finding a molecular representation that could uniquely describe a given molecule. For this purpose, we originally tried features from a method called "Geometric Scattering for Graphs",[45] which provides truly index-invariant features by "scattering" atomic signals over the molecular graph structure, and storing summations of different moments of the scattered signals as features. We had shown previously that this approach could predict partition coefficients for small molecules to high accuracy.[46,47] We attempted to use these GSG features by themselves, as well as GSG features computed with the Behler−Parrinello symmetry functions as signals, and none of these feature sets were able to generate proper models for Flexible Topology simulations. In these cases, we found that minimizing the loss function was not sufficient to achieve the desired ligand structures. In other words, many structures were possible that were far from the target ligand structure, but still reproduced the features exactly. The current implementation, which directly compares atomic feature values after an on-the-fly Hungarian algorithm mapping step, was found to be superior to the graph scattering approach. A limitation of the approach used here is that a single set of target features enforces a single rigid conformation of the ligand. While there are some ways to alleviate this, such as including multiple sets of target features corresponding to different local minima, this approach is likely unsuitable for more flexible ligands.

Sets of parameters were identified that consistently achieved high assembly accuracy for ligands up to 50 atoms in size. Some key elements of the approach that enabled the assembly of the larger ligands were: (1) long assignment intervals, (2) second-order restraints, and (3) strong MLForce scales. We found that increasing the strength of the second-order restraints led to instabilities in the integrator, especially when combined with shorter assignment intervals. This could possibly be alleviated with a better assignment algorithm that takes the energy of the second-order assignments into account. Unfortunately, such an assignment algorithm could be far more costly than the Hungarian algorithm. One approach could be to use the Hungarian algorithm as an initial assignment, which is refined by a set of Monte Carlo assignment swaps that are accepted or rejected using the full restraint energy (including the second-order assignments). As this only needs to be run periodically, this could be introduced with a reasonable impact on performance. Another possible improvement would be to learn a set of scaling parameters for the loss function

$$L_{scaled} = \sum_i w_i (F^i_{GP} - F^i_T)^2 \tag{11}$$

where w_i is the scale factor for feature i. This would allow us to "train" the loss function for optimal performance in assembly simulations. We reserve these lines of investigation for future work.

As the s parameter gradually increases the strength of the MLForce restraint, we can measure the work performed by the restraint on the system. These trajectories can be thought of as nonequilibrium processes in which the system is driven from

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098

the disassembled state ($s = 0$) to the assembled state ($s = s_{max}$). The total work applied over the course of these trajectories can be related to their free energy difference using the Jarzynski equality.[48] By calculating this free energy difference—ΔG_A, the free energy of assembly—in different environments we can calculate transfer free energies between different media, or even binding free energies. This has the potential to offer some gains in efficiency compared to alchemical approaches for determining the absolute binding free energy, as the disordered state allows for rapid sampling of different bound orientations and solvation configurations within the binding pocket. In addition, the perturbation between the disordered and ordered state could be smaller than the standard alchemical perturbation between the $\lambda = 1$ state (full pocket) and $\lambda = 0$ state (empty pocket). Practical concerns, such as how many trajectories are required and how slowly the restraints can be turned on, will dictate whether this approach will be competitive compared to other alchemical approaches.

Another potential application of this method is to the prediction of ligand-bound poses. By assembling the ligand in the binding pocket of a protein of interest, the assembled conformation constitutes a prediction of the ligand binding pose. This could offer advantages compared with approaches like docking, as it can explicitly simulate ligand-induced fit and collective relaxation of water molecules in a binding site. Again, it is likely that a slower introduction of the restraints would be necessary in order to allow the binding pocket and water molecules to adapt to the presence of the ligand. Here we find that successful assembly can be achieved in as little as 60 ps. To observe meaningful conformational adaptation in a protein binding pocket we would likely need to slow this down considerably.

A long-term goal of this work is to build a method that can be used to run dynamics in a joint chemical-conformational space that is entirely continuous. This would allow for the exploration of chemical space while using molecular dynamics (MD) to model ligand-induced conformational changes in the receptor as well as water molecules' effects on ligand-receptor interactions. For large datasets, such a method would have the potential for dramatic gains in efficiency. To illustrate this, it is helpful to consider the example of a protein searching its conformational landscape to find its folded state. Levinthal's paradox states that there are an enormous number of possible protein conformations (10^{300})[49] and if all of the conformations of protein are sampled one by one at a rate of 1 conformation per millisecond, then it would take the universe's age to find its folded conformation. However, in nature, protein folding does not require sampling all of the high-energy conformations; the protein is automatically drawn toward low-energy conformations due to the funnel-shaped nature of the energy landscape.[50] The same reasoning is valid for a coupled chemical-conformational landscape approach as its search space will also be funnel-shaped with the lowest-energy target molecules at its bottom. In this way, we can simply avoid sampling all of the high-energy ligands and conformations and be gradually drawn toward the best chemicals and structures with each simulation step.

Some modifications to the algorithm are needed before these applications can be realized. The loss function used here is defined using only a single set of target features. One modification is to employ a loss function with multiple minima that represent either different ligands

$$L = \min_{i \in I}(|M(F_{GP}) - F_{T,i}| + \varsigma_i) \tag{12}$$

where only one set of target features $F_{T,i}$ in a (possibly large) set (I) is used to determine L and an offset for each target (ς_i) can be used to balance their relative probabilities. For instance, if we are running a long simulation, or a large ensemble of simulations, ς_i could be progressively increased for each frame that ligand is correctly assembled. This would discourage repeated assembly of the same ligands and result in a more diverse ensemble of hits. A practical challenge to overcome is handling mismatches in the number of atoms between ligands. The total number of ghost particles needs to be larger or equal to the number of atoms in the largest ligand in I. To match smaller ligands, the λ values of the unmatched atoms should go to zero. The value with which unmatched atoms contribute to the loss for a given ligand should be nonzero and tuned empirically to ensure sampling is as even as possible across all ligand sizes.

There are many other methods emerging for ligand design that use the structure of the binding site as input. Flexible Topology presents a means of incorporating these strategies into a physics-based simulation framework. For instance, generative models,[51−53] informed by the current structure of the binding pocket, and/or the positions and attributes of the ghost particles, could propose new target ligands on the fly. Another approach is to replace L with a general machine-learned potential from a diffusion model.[54,55] This could be trained using many conformations of a single ligand, or with conformations from a large ligand set.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.3c00409.

Scatter plot of minimum loss per particle vs. minimum RMSD (Figure S1) and average loss per particle for varying MLForce Scale values (Figure S2) (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Alex Dickson − *Department of Biochemistry & Molecular Biology, Michigan State University, East Lansing, Michigan 48824, United States; Department of Computational Mathematics, Science & Engineering, Michigan State University, East Lansing, Michigan 48824, United States;* orcid.org/0000-0002-9640-1380; Email: alexrd@msu.edu

Authors

Nazanin Donyapour − *Department of Computational Mathematics, Science & Engineering, Michigan State University, East Lansing, Michigan 48824, United States*

Fatemeh Fathi Niazi − *Department of Computational Mathematics, Science & Engineering, Michigan State University, East Lansing, Michigan 48824, United States*

Nicole M. Roussey − *Department of Biochemistry & Molecular Biology, Michigan State University, East Lansing, Michigan 48824, United States*

Samik Bose − *Department of Biochemistry & Molecular Biology, Michigan State University, East Lansing, Michigan 48824, United States*

Complete contact information is available at:

https://pubs.acs.org/10.1021/acs.jctc.3c00409

Author Contributions

A.D. designed the project; N.D. implemented the method; all authors collected and analyzed the data; A.D. and N.D. prepared the manuscript; and all authors provided edits and feedback.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by R01GM130794 from the National Institutes of Health and by DMS 1761320 from the National Science Foundation.

■ REFERENCES

(1) Liu, P.; Kim, B.; a Friesner, R.; Berne, B. J. Replica exchange with solute tempering: a method for sampling biological systems in explicit water. *Proc. Nat. Acad. Sci. U.S.A.* **2005**, *102*, 13749−13754.
(2) Cérou, F.; Guyader, A. Adaptive Multilevel Splitting for Rare Event Analysis. *Stochastic Anal. Appl.* **2007**, *25*, 417−443.
(3) Tiwary, P.; Parrinello, M. From Metadynamics to Dynamics. *Phys. Rev. Lett.* **2013**, *111*, No. 230602.
(4) Dickson, A.; III, C. L. B. WExplore: Hierarchical exploration of high-dimensional spaces using the weighted ensemble algorithm. *J. Phys. Chem. B* **2014**, *118*, 3532−3542.
(5) Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J. Chem. Theory Comput.* **2015**, *11*, 5747−5757.
(6) Ahn, S. H.; Grate, J. W.; Darve, E. F. Efficiently sampling conformations and pathways using the concurrent adaptive sampling (CAS) algorithm. *J. Chem. Phys.* **2017**, *147*, No. 074115.
(7) Chen, W.; Tan, A. R.; Ferguson, A. L. Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.* **2018**, *149*, No. 072312.
(8) Donyapour, N.; Roussey, N. M.; Dickson, A. REVO: Resampling of ensembles by variation optimization. *J. Chem. Phys.* **2019**, *150*, No. 244112.
(9) Zhang, J.; Gong, H. Frontier expansion sampling: a method to accelerate conformational search by identifying novel seed structures for restart. *J. Chem. Theory Comput.* **2020**, *16*, 4813−4821.
(10) Bonati, L.; Rizzi, V.; Parrinello, M. Data-Driven Collective Variables for Enhanced Sampling. *J. Phys. Chem. Lett.* **2020**, *11*, 2998−3004.
(11) Wan, H.; Voelz, V. A. Adaptive Markov state model estimation using short reseeding trajectories. *J. Chem. Phys.* **2020**, *152*, No. 024103.
(12) Votapka, L. W.; Stokely, A. M.; Ojha, A. A.; Amaro, R. E. SEEKR2: Versatile Multiscale Milestoning Utilizing the OpenMM Molecular Dynamics Engine. *J. Chem. Inf. Model.* **2022**, *62*, 3253−3262.
(13) Reymond, J.-L. The Chemical Space Project. *Acc. Chem. Res.* **2015**, *48*, 722−730.
(14) Frimurer, T. M.; Peters, G. H.; Iversen, L. F.; Andersen, H. S.; Møller, N. P. H.; Olsen, O. H. Ligand-induced conformational changes: Improved predictions of ligand binding conformations and affinities. *Biophys. J.* **2003**, *84*, 2273−2281.
(15) Provasi, D.; Artacho, M. C.; Negri, A.; Mobarec, J. C.; Filizola, M. Ligand-induced modulation of the free-energy landscape of G protein-coupled receptors explored by adaptive biasing techniques. *PLoS Comp. Bio.* **2011**, *7*, No. e1002193.
(16) Bai, Q.; Pérez-Sánchez, H.; Zhang, Y.; Shao, Y.; Shi, D.; Liu, H.; Yao, X. Ligand induced change of beta2 adrenergic receptor from active to inactive conformation and its implication for the closed/open state of the water channel: insight from molecular dynamics simulation, free energy calculation and Markov state model analysis. *Phys. Chem. Chem. Phys.* **2014**, *16*, 15874−15885.
(17) Altucci, L.; Leibowitz, M. D.; Ogilvie, K. M.; de Lera, A. R.; Gronemeyer, H. RAR and RXR modulation in cancer and metabolic disease. *Nat. Rev. Drug Discovery* **2007**, *6*, 793−810.
(18) Pioszak, A. A.; Hay, D. L. RAMPs as allosteric modulators of the calcitonin and calcitonin-like class B G protein-coupled receptors. *Adv. Pharmacol.* **2020**, *88*, 115−141.
(19) Tembre, B. L.; Mc Cammon, J. A. Ligand-receptor interactions. *Comput. Chem.* **1984**, *8*, 281−283.
(20) Khandogin, J.; Brooks, C. L. Constant pH molecular dynamics with proton tautomerism. *Biophys. J.* **2005**, *89*, 141−157.
(21) Chen, W.; Morrow, B. H.; Shi, C.; Shen, J. K. Recent development and application of constant pH molecular dynamics. *Mol. Simul.* **2014**, *40*, 830−838.
(22) Radak, B. K.; Chipot, C.; Suh, D.; Jo, S.; Jiang, W.; Phillips, J. C.; Schulten, K.; Roux, B. Constant-pH molecular dynamics simulations for large biomolecular systems. *J. Chem. Theory Comput.* **2017**, *13*, 5933−5944.
(23) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. Guidelines for the analysis of free energy calculations. *J. Comp.-Aided Mol. Des.* **2015**, *29*, 397−411.
(24) Mobley, D. L.; Graves, A. P.; Chodera, J. D.; McReynolds, A. C.; Shoichet, B. K.; Dill, K. A. Predicting absolute ligand binding free energies to a simple model site. *J. Mol. Bio.* **2007**, *371*, 1118−1134.
(25) Chodera, J. D.; Mobley, D. L.; Shirts, M. R.; Dixon, R. W.; Branson, K.; Pande, V. S. Alchemical free energy methods for drug discovery: progress and challenges. *Curr. Opin. Struct. Bio.* **2011**, *21*, 150−160.
(26) Kong, X.; Brooks III, C. L. λ-dynamics: A new approach to free energy calculations. *J. Chem. Phys.* **1996**, *105*, 2414−2423.
(27) Guo, Z.; Durkin, J.; Fischmann, T.; Ingram, R.; Prongay, A.; Zhang, R.; Madison, V. Application of the λ-dynamics method to evaluate the relative binding free energies of inhibitors to HCV protease. *J. Med. Chem.* **2003**, *46*, 5360−5364.
(28) Guo, Z.; Brooks, C. L. Rapid screening of binding affinities: application of the λ-dynamics method to a trypsin-inhibitor system. *J. Am. Chem. Soc.* **1998**, *120*, 1920−1921.
(29) Banba, S.; Guo, Z.; Brooks, C. L. Efficient sampling of ligand orientations and conformations in free energy calculations using the λ-dynamics method. *J. Phys. Chem. B* **2000**, *104*, 6903−6910.
(30) Banba, S.; Brooks III, C. L. Free energy screening of small ligands binding to an artificial protein cavity. *J. Chem. Phys.* **2000**, *113*, 3423−3433.
(31) Damodaran, K.; Banba, S.; Brooks, C. L. Application of multiple topology λ-dynamics to a host- guest system: β-cyclodextrin with substituted benzenes. *J. Phys. Chem. B* **2001**, *105*, 9316−9322.
(32) Knight, J. L.; Brooks, C. L. Multisite lambda dynamics for simulated structure-activity relationship studies. *J. Chem. Theory Comput.* **2011**, *7*, 2728−2739.
(33) Raman, E. P.; Paul, T. J.; Hayes, R. L.; III, C. L. B. Automated, Accurate, and Scalable Relative Protein−Ligand Binding Free-Energy Calculations Using Lambda Dynamics. *J. Chem. Theory Comput.* **2020**, *16*, 7895−7914.
(34) Robo, M.; Hayes, R.; Ding, X.; Pulawski, B.; Vilseck, J. Achieving Rapid Free Energy Estimates from lambda-Dynamics with Bias Updated Gibbs Sampling, PREPRINT, 2022 https://doi.org/10.21203/rs.3.rs-1551844/v1.
(35) Huang, J.; MacKerell Jr, A. D. CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *J. Comp. Chem.* **2013**, *34*, 2135−2145.
(36) Donyapour, N.; Roussey, N. M.; Dickson, A.; Fathi Niazi, F.; Bose, S. MLForce. https://github.com/ADicksonLab/mlforce.
(37) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, No. 146401.
(38) Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logistics Q.* **1955**, *2*, 83−97.

(39) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comp. Bio.* **2017**, *13*, No. e1005659.
(40) Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E. TorchANI: a free and open source PyTorch-based deep learning implementation of the ANI neural network potentials. *J. Chem. Inf. Model.* **2020**, *60*, 3408−3415.
(41) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. *Advances in Neural Information Processing Systems 32*; Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché Buc, F.; Fox, E.; Garnett, R., Eds.; Curran Associates, Inc., 2019; pp 8024−8035.
(42) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, *8*, 3192−3203.
(43) Donyapour, N.; Roussey, N. M.; Dickson, A.; Fathi Niazi, F. FlexibleTopology. https://github.com/ADicksonLab/flexibletopology.
(44) Korshunova, M.; Ginsburg, B.; Tropsha, A.; Isayev, O. OpenChem: A Deep Learning Toolkit for Computational Chemistry and Drug Design. *J. Chem. Inf. Model.* **2021**, *61*, 7−13.
(45) Gao, F.; Wolf, G.; Hirn, M.*Geometric Scattering for Graph Data Analysis*, International Conference on Machine Learning, 2019; pp 2122−2131.
(46) Donyapour, N.; Hirn, M.; Dickson, A. ClassicalGSG: Prediction of log P using classical molecular force fields and geometric scattering for graphs. *J. Comput. Chem.* **2021**, *42*, 1006−1017.
(47) Donyapour, N.; Dickson, A. Predicting partition coefficients for the SAMPL7 physical property challenge using the ClassicalGSG method. *J. Comp.-Aided Mol. Des.* **2021**, *35*, 819−830.
(48) Jarzynski, C. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.* **1997**, *78*, 2690.
(49) Levinthal, C. *Mossbauer Spectroscopy in Biological Systems*, Proceedings of a meeting held at Allerton House; Debrunner, P.; Tsibris, J. C. M.; Munck, E., Eds.; University of Illinois Press: Urbana, IL, 1969.
(50) Leopold, P. E.; Montal, M.; Onuchic, J. N. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proc. Natl. Acad. Sci.* **1992**, *89*, 8721−8725.
(51) Luo, S.; Guan, J.; Ma, J.; Peng, J. A 3D Generative Model for Structure-Based Drug Design. 2021, pp 6229−6239.
(52) Wang, M.; Hsieh, C.-Y.; Wang, J.; Wang, D.; Weng, G.; Shen, C.; Yao, X.; Bing, Z.; Li, H.; Cao, D.; Hou, T. RELATION: A Deep Generative Model for Structure-Based De Novo Drug Design. *J. Med. Chem.* **2022**, *65*, 9478−9492.
(53) Adams, K.; Coley, C. W. Equivariant Shape-Conditioned Generation of 3D Molecules for Ligand-Based Drug Design, arXiv, 2022 http://arxiv.org/abs/2210.04893.
(54) Salimans, T.; Ho, J. Should EBMs model the energy or the score? Energy Based Models Workshop - ICLR 2021. 2021.
(55) Arts, M.; Satorras, V. G.; Huang, C.-W.; Zuegner, D.; Federici, M.; Clementi, C.; Noé, F.; Pinsler, R.; van den Berg, R. Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics. 2023.

https://doi.org/10.1021/acs.jctc.3c00409
J. Chem. Theory Comput. 2023, 19, 5088−5098