Rigid Body Flows for Sampling Molecular Crystal Structures

Jonas Köhler [*1 2] **Michele Invernizzi** [*2] **Pim de Haan** [3 4] **Frank Noé** [1 2 5 6]

Abstract

Normalizing flows (NF) are a class of powerful generative models that have gained popularity in recent years due to their ability to model complex distributions with high flexibility and expressiveness. In this work, we introduce a new type of normalizing flow that is tailored for modeling positions and orientations of multiple objects in three-dimensional space, such as molecules in a crystal. Our approach is based on two key ideas: first, we define smooth and expressive flows on the group of unit quaternions, which allows us to capture the continuous rotational motion of rigid bodies; second, we use the double cover property of unit quaternions to define a proper density on the rotation group. This ensures that our model can be trained using standard likelihood-based methods or variational inference with respect to a thermodynamic target density. We evaluate the method by training Boltzmann generators for two molecular examples, namely the multi-modal density of a tetrahedral system in an external field and the ice XI phase in the TIP4P water model. Our flows can be combined with flows operating on the internal degrees of freedom of molecules, and constitute an important step towards the modeling of distributions of many interacting molecules.

1. Introduction

Normalizing flows (NF) (Tabak et al., 2010; Rezende & Mohamed, 2015; Papamakarios et al., 2021) are popular deep learning generative models, that have been applied to the physical sciences in a variety of ways, such as for sampling lattice models (Nicoli et al., 2020; 2021; Li & Wang, 2018; Boyda et al., 2021; Albergo et al., 2019), approximating the equilibrium density of molecular systems (Noé et al., 2019; Köhler et al., 2020; Wu et al., 2020; Xu et al., 2021), and estimating free energy differences (Wirnsberger et al., 2020; Ding & Zhang, 2021). This success is motivated by the fact that, contrary to other generative models, NF are built to provide an efficient reweighting scheme that allows for exact sampling from a given energy-based probability distribution. In this work we present a novel NF architecture that is particularly suited to sample molecular crystals, i.e. periodic systems where multiple copies of the same molecule are arranged in a lattice structure. Molecular crystals are of key importance for several applications, ranging from the pharmaceutical industry to solar energy production (Bernstein, 2020). For this reason, there is great interest in developing efficient and effective methods to predict the physical properties of molecular crystals through computer simulations rather than expensive laboratory experiments.

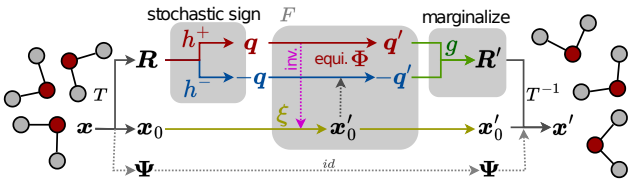

Figure 1. Flow on SO(3) via the S^3 double cover: we first transform a system of Cartesian coordinates $\boldsymbol{x}$ via T into the triple $(\boldsymbol{x}_0, \boldsymbol{R}, \boldsymbol{\Psi})$, containing the global translation $\boldsymbol{x}_0$, the global rotation $\boldsymbol{R}$ and its fixed inner degrees of freedom $\boldsymbol{\Psi}$ (e.g., bond lengths and inner angle of water molecules). While keeping $\boldsymbol{\Psi}$ fixed we transform the poses as follows: first map $\boldsymbol{R}$ onto one of the two representing quaternions $\boldsymbol{q}$ or $-\boldsymbol{q}$ stochastically (here we picked $\boldsymbol{q}$ as indicated in red). Then transform $(\boldsymbol{x}_0, \boldsymbol{q})$ into $(\boldsymbol{x}_0', \boldsymbol{q}')$ using a sign-flip equivariant coupling flow F made of position updates ξ and rotation updates Φ. As the double cover projection g maps both $\boldsymbol{q}'$ and $-\boldsymbol{q}'$ to the same rotation element $\boldsymbol{R}'$, transforming back corresponds to a simple marginalization over both stochastic paths. We finally obtain $\boldsymbol{x}'$ using the inverse rigid body transform T^{-1}.

Normalizing flows for equilibrium sampling and free-energy difference estimation In this work, we focus on methods that have a primary application in the field of Boltzmann Generators (BG) (Noé et al., 2019). BGs are gener-

[*]Equal contribution [1]Microsoft Research AI4Science [2]Freie Universität Berlin, Department of Mathematics and Computer Science [3]Qualcomm AI Research, an initiative from Qualcomm Technologies, Inc. [4]University of Amsterdam [5]Freie Universität Berlin, Department of Physics [6]Rice University, Department of Chemistry. Correspondence to: Frank, Noé <franknoe@microsoft.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

ative models that are trained to sample conformations of molecules in equilibrium. These follow a Boltzmann-type distribution, $\mu(\boldsymbol{x}) \propto \exp(-u(\boldsymbol{x}))$. Here u is the dimensionless potential defined by the molecular system and the thermodynamic state in which it is simulated, such as the canonical ensemble at a certain temperature. BGs are primarily implemented using NFs and as such can be trained using a combination of maximum likelihood estimation on potentially biased data obtained from molecular dynamics (MD) simulations, and energy-based training via the reverse Kullback-Leibler (KL) divergence. Once trained, BGs can be used for importance sampling (Noé et al., 2019; Müller et al., 2019), as efficient proposals in Markov chain Monte Carlo (MCMC) applications (Sbailò et al., 2021; Gabrié et al., 2022), or as teacher models when learning coarse-grained MD force-fields (Köhler et al., 2023). An important advantage of BGs over traditional methods such as MD and MCMC, is that the latter struggle to efficiently sample systems characterized by long-lived metastable states separated by very low-probability transition regions. Such multistable systems are ubiquitous, e.g. chemical reactions, conformational rearrangements in biomolecules, phase transitions in materials. One of the most important property for these systems, is the relative stability of their metastable states, i.e. their free energy difference. Boltzmann generators and NF can be used to directly connect such metastable states and compute the free energy difference (Wirnsberger et al., 2020; Rizzi et al., 2021; Invernizzi et al., 2022).

Building flows on natural molecular representations Despite their potential, previous flow architectures for molecular systems have severe limitations. Most importantly, many high-fidelity models rely on representations that are either non-scalable (e.g., global internal coordinates (Köhler et al., 2021; Köhler et al., 2023; Invernizzi et al., 2022)), non-transferable (e.g., principal components (Noé et al., 2019)), or unnatural (e.g., splits between Cartesian axes (Wirnsberger et al., 2020; 2022)). Equivariant all-atom representations, while more principled, require computationally intensive and approximate methods such as solving neural ODEs (Köhler et al., 2020; Garcia Satorras et al., 2021) and can be challenging to scale and integrate with energy-based training.

A more natural and scalable representation would place atoms relative to the orientation and position of a chemical entity such as the molecule or residue, thus separating intermolecular degrees of freedom and intra-molecular placement of atoms relative to the pose. This becomes especially important in solvated systems and molecular crystals, where the most interesting emergent properties result from intermolecular interactions, whereas intra-molecular degrees of freedom vary often predictably.

Additionally, when simulating such systems, it is common practice to fix the stiffest internal degrees of freedom, typically inter-atomic bonds and angles, as it allows for larger integration time steps and thus faster mixing without giving up much accuracy. Yet, if rigid residues are present in a simulation, the molecular density μ becomes non-singular only for a sub-manifold of the full space. Thus, any NF approach modeling all degrees of freedom can neither be reweighted against such a singular target density nor can it be trained by minimizing the reverse KL divergence.

Transforming the molecular pose independently from inner degrees of freedom requires a physically meaningful normalizing flow architecture on the pose manifold that scales to systems composed of many interacting poses.

Contributions Here, we present such an approach to designing normalizing flows for sampling the joint distribution over positions and orientations of systems composed of many molecules. As handling the internal degrees of freedom separately was extensively studied in prior work, such as (Köhler et al., 2021), we focus on the important limit case of purely rigid bodies in this work. Our flow architecture is ideal for molecular simulation applications due to its unique traits, including:

- They prescribe fully smooth densities on the rigid body sub-manifold. The smoothness of the flow density has shown to be critical for faithful modeling of physical force fields (Köhler et al., 2021; Köhler et al., 2023).
- They are compatible with permutation equivariant architectures, such as used in Wirnsberger et al. (2022). This feature has been shown to be critical when scaling flows to larger bulk systems, e.g., extended crystals or water boxes.
- External pose and inner degrees of freedom are treated independently. As such they are fully compatible with prior work focusing on modeling the internal degrees of freedom separately.

As part of the method we further contribute two new smooth flow architectures for the rotation manifold $SO(3)$, namely symmetrized Moebius transformations and projective convex gradient maps. We finally demonstrate the efficacy and efficiency of the method by sampling the multi-modal density of a tetrahedral body in an external field, as well as sampling ice XI crystals following the TIP4P water model at different sizes and temperatures with high accuracy.

2. Related Work

Normalizing flows on manifolds Normalizing flows on manifolds have been extensively studied for Riemannian geometry, e.g., in the form of convex potential flows (Cohen

et al., 2021; Rezende & Racanière, 2021) or neural ODEs (Chen et al., 2018) on manifolds (Lou et al., 2020; Katsman et al., 2021; Mathieu & Nickel, 2020; Falorsi, 2021; Ben-Hamu et al., 2022). Approaches to smooth coupling flows on non-trivial manifolds, like tori and spheres, were discussed in Rezende et al. (2020); Köhler et al. (2021). Beyond that there exist approaches to non-smooth normalizing flows on Riemannian submanifolds via charts (Gemici et al., 2016; Kalatzis et al., 2021). Using the double cover with normalizing flows to estimate densities of single poses in the context of computer vision was concurrently discussed in the work of Liu et al. (2023). We discuss the relation of this concurrent work to the present work in Sec. 3.2.

Density estimation on SO(3) Beyond flows other methods for neural density estimation on $SO(3)$ have been proposed for domains outside of molecular physics: Falorsi et al. (2019) discussed using the exponential map to push-forward densities on the Lie-algebra $\mathfrak{so}(3)$ to $SO(3)$. Furthermore, Murphy et al. (2021) proposed single pose estimation using the double cover via an implicit neural representation on S^3.

Sampling equilibrium structures with normalizing flows Normalizing flows for sampling molecular systems were studied in the context of importance sampling (Wu et al., 2020; Köhler et al., 2020; Dibak et al., 2021; Köhler et al., 2021; Midgley et al., 2022) and estimating free energy differences (Noé et al., 2019; Wirnsberger et al., 2020; Ding & Zhang, 2021; Rizzi et al., 2021; Wirnsberger et al., 2022; Invernizzi et al., 2022; Ahmad & Cai, 2022; Coretti et al., 2022). Furthermore, Garcia Satorras et al. (2021) used NF for generating conformations across molecular space, however only focusing on density estimation without explicit treatment of the thermodynamics.

Sampling molecular crystals without machine learning Established methods to sample molecular crystals typically rely on molecular dynamics or Markov chain Monte Carlo simulations (Frenkel & Smit, 2001). Several protocols have been proposed to compute free energy difference and phase diagrams for molecular crystals, one of the most popular being thermodynamic integration (Frenkel & Ladd, 1984; Vega et al., 2008). Relevant to our purposes is the targeted free energy perturbation method (Jarzynski, 2002) that has been combined with the multistate Bennett acceptance ratio (MBAR) (Shirts & Chodera, 2008) to compute crystal free energies (Schieber et al., 2018; Schieber & Shirts, 2019).

3. Theory & Method

In molecular equilibrium sampling we are provided with a (dimensionless) potential function

$$u(\boldsymbol{x})\colon \mathbb{R}^{N\times 3} \to \mathbb{R} \tag{1}$$

describing the probability density of samples $\mu(\boldsymbol{x})$ in thermodynamic equilibrium, via the relation

$$\mu(\boldsymbol{x}) = \frac{\exp(-u(\boldsymbol{x}))}{Z}, \tag{2}$$

where Z is the unknown normalizing constant (partition function) of the system.

In the more general case, we further assume a parametric ensemble of potentials u_α with corresponding densities μ_α, where each α corresponds to a different thermodynamic state, e.g., pressure or temperature.

Primary goals are now

1. drawing asymptotically unbiased i.i.d. samples from μ from which we can estimate expectation values of downstream observables, as well as,
2. estimating the log-ratio

$$\Delta F = -\log\left(Z_{\alpha_1}/Z_{\alpha_0}\right) \tag{3}$$

between the partition functions Z_{α_0} and Z_{α_1} of two thermodynamic states α_0 and α_1. This quantity is also called *free energy difference* between u_{α_0} and u_{α_1} and is an important measure telling which state is more stable among different thermodynamic conditions.

Previous work (Noé et al., 2019; Wirnsberger et al., 2020) has shown that NFs are a natural choice for such tasks, as they allow to formulate the sampling problem relative to a tractable base density, while providing asymptotic guarantees on unbiasedness.

NF approximate a target density μ by a parametric diffeomorphic map, $\Phi(\cdot;\boldsymbol{\theta})$, that transforms samples from a base density, $\boldsymbol{z} \sim p_0(\boldsymbol{z})$, into samples that follow the push-forward density,

$$\Phi_*(p_0)(\boldsymbol{x};\boldsymbol{\theta}) := p_0\left(\Phi^{-1}(\boldsymbol{x};\boldsymbol{\theta})\right)\left|\boldsymbol{J}_\Phi^{-1}(\boldsymbol{x};\boldsymbol{\theta})\right| \tag{4}$$

In standard applications, like density estimation on images, the parameters $\boldsymbol{\theta}$ are learned by minimizing the negative log-likelihood on data

$$\boldsymbol{\theta}_{ML} = \arg\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mu(\boldsymbol{x})}\left[-\log \Phi_*(p_0)(\boldsymbol{x};\boldsymbol{\theta})\right]. \tag{5}$$

In the molecular sampling setup, however, samples from $\mu(\boldsymbol{x})$ are usually sparse and biased (e.g. when obtained from non-converged MD simulations). As such, training can combine biased likelihood training with minimizing the reverse Kullback-Leibler divergence

$$\boldsymbol{\theta}_{KL} = \arg\min_{\theta} D_{KL}\left[\Phi_*(p_0)(\cdot;\boldsymbol{\theta})\|\mu(\cdot)\right]. \tag{6}$$

If the base density p_0 is given by a simple density with closed-form sampling algorithm, e.g., an isotropic Gaussian, these model can be turned into asymptotically unbiased independence samplers using reweighing techniques, such as self-normalized importance sampling (Noé et al., 2019). These models where coined *Boltzmann generators* in prior work and can be used to tackle goal 1.

Alternatively, we could consider $p_0 = \mu_{\alpha_0}$, for some reference potential u_{α_0} and learn the mapping to any other potential u_{α_1} via reverse KL minimization. This leads to the method of *learned free energy perturbation* (LFEP) (Wirnsberger et al., 2020) and allows to directly estimate

$$\Delta F \leq D_{KL}\left[\Phi_*(\mu_{\alpha_0})(\cdot;\boldsymbol{\theta})\|\mu_{\alpha_1}(\cdot)\right] \tag{7}$$

from above. In practice one can, e.g., run MD on μ_{α_0} to obtain samples, train the flow using loss 6, and then estimate the bound 7.

Normalizing flows for rigid bodies We now explain how this framework can be used when studying systems composed of rigid bodies

Let us assume we have a system $\boldsymbol{X} \in \mathbb{R}^{N\times K\times 3}$ consisting of N bodies, each consisting of K beads in $\mathbb{R}^3$. In the present model, we assume that these bodies are rigid, i.e., we only sample the joint translation or rotation of the K beads. However, combining this rigid body flow with a flow operating on internal degrees of freedom is conceptually straightforward.

Each rigid body $\boldsymbol{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{K-1})$ can equivalently be described as a triple $(\boldsymbol{x}_0, \boldsymbol{R}, \boldsymbol{\Psi})$ of the position $\boldsymbol{x}_0 \in \mathbb{R}^3$ of the first bead, a rotation matrix $\boldsymbol{R} \in SO(3) \subset \mathbb{R}^{3\times 3}$, and $3K-6$ inner degrees of freedom $\boldsymbol{\Psi}$. An intuitive approach to modeling a diffeomorphism in this representation is keeping $\boldsymbol{\Psi}$ fixed and only describing transformations of $(\boldsymbol{x}_0, \boldsymbol{R})$.

While modeling flows on $\mathbb{R}^3$ is clear, modeling smooth normalizing flows directly on $SO(3)$ is challenging and has pros and cons depending on the representation. Rotation matrices are the natural way to handle rotational degrees of freedom, but designing expressive normalizing flows can be difficult due to the orthonormality and sign constraint. Euler angles on the other hand show the gimbal lock phenomenon leading to a non-smooth representation. They furthermore act nonlinearly and induce a volume change. Another drawback of working on $SO(3)$ directly is that equivariance under rotations is very difficult to incorporate into the map. As shown, e.g., in Köhler et al. (2020); Garcia Satorras et al. (2021) this can become critical when scaling flows to larger physical systems. While there exist intrinsic manifold approaches, such as Falorsi (2021); Mathieu & Nickel (2020); Lou et al. (2020), those require expensive and possibly inexact numerical integration methods and as such are hard to scale. Furthermore, computing their exact density is usually avoided via stochastic approximation of the divergence term which in practice is not sufficient for accurate reweighting of molecular systems (Köhler et al., 2020).

Here we give a formulation of normalizing flows for rigid bodies that are smooth, fast to compute and invert, compatible with equivariance constraints, and provide a tractable exact density.

3.1. Flows on $SO(3)$ via the $S^3 \to SO(3)$ double cover.

Instead of working on $SO(3)$ directly, we can also model rotations via the group of *unit quaternions*, also known as SU(2), i.e., the set of unit vectors $\boldsymbol{q} \in S^3 \subset \mathbb{R}^4$ equipped with the quaternion product. We denote the quaternion product between two quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ as $\boldsymbol{q}_1 \odot \boldsymbol{q}_2$, and the conjugation of $\boldsymbol{q}$ as $\boldsymbol{q}^*$. Let $\iota\colon \mathbb{R}^3 \hookrightarrow \mathbb{R}^4$ be the canonical embedding where we embed a point $\boldsymbol{x} \in \mathbb{R}^3$ as a purely imaginary quaternion $(x_0, x_1, x_2, 0)$, and $\pi\colon \mathbb{R}^4 \twoheadrightarrow \mathbb{R}^3$ be the corresponding projection, such that $\pi \circ \iota = \mathrm{id}$. For any $\boldsymbol{q} \in S^3$ the map $\boldsymbol{R}_{\boldsymbol{q}}\colon \mathbb{R}^3 \to \mathbb{R}^3, \boldsymbol{x} \mapsto \boldsymbol{q}.\boldsymbol{x} := \pi\left(\boldsymbol{q} \odot \iota(\boldsymbol{x}) \odot \boldsymbol{q}^*\right)$ is a rotation of $\boldsymbol{x}$ around the origin. This defines a smooth map $g\colon S^3 \to SO(3), \boldsymbol{q} \mapsto \boldsymbol{R}_{\boldsymbol{q}}$. Furthermore, g is surjective and as such, each rotation $\boldsymbol{R}$ can be represented by some $\boldsymbol{q} \in g^{-1}(\boldsymbol{R})$. However, g is not injective as we have $\boldsymbol{q}.\boldsymbol{x} = (-\boldsymbol{q}).\boldsymbol{x}$. In fact, g forms a covering map, i.e., for each $\boldsymbol{R} \in SO(3)$ there is an open neighborhood $U_{\boldsymbol{R}}$, such that $g^{-1}(U_{\boldsymbol{R}}) \approx U_{\boldsymbol{R}} \times \mathbb{Z}_2$. Furthermore for each $\boldsymbol{R}$ there are locally defined smooth functions $h^+_{\boldsymbol{R}}, h^-_{\boldsymbol{R}}\colon U_{\boldsymbol{R}} \to S^3$, such that, $g \circ h^+_{\boldsymbol{R}} = g \circ h^-_{\boldsymbol{R}} = \mathrm{id}_{U_{\boldsymbol{R}}}$ and $h^+_{\boldsymbol{R}} = -h^-_{\boldsymbol{R}}$. Abusing notation we will abbreviate $\boldsymbol{q}(\boldsymbol{R}) := h^+_{\boldsymbol{R}}(\boldsymbol{R})$ and $-\boldsymbol{q}(\boldsymbol{R}) := h^-_{\boldsymbol{R}}(\boldsymbol{R})$ in the following discussion.

Stochastic paths over the double cover We can leverage this smooth double cover to design smooth flows for rigid bodies in the following way (see Fig. 1):

- We first transform $\boldsymbol{x}$ into its rigid representation $(\boldsymbol{x}_0, \boldsymbol{R}, \boldsymbol{\Psi})$. This can be done, e.g., by computing the pose $(\boldsymbol{x}_0, \boldsymbol{R})$, removing it from $\boldsymbol{x}$, and then computing $\boldsymbol{\Psi}$ in this standard frame. We keep $\boldsymbol{\Psi}$ fixed and write the transformation relative to $\boldsymbol{\Psi}$ as $T_{\boldsymbol{\Psi}}\colon \boldsymbol{x} \mapsto (\boldsymbol{x}_0, \boldsymbol{R})$.
- We then stochastically embed $\boldsymbol{R}$ as either $\boldsymbol{q}(\boldsymbol{R})$ or

$-\boldsymbol{q}(\boldsymbol{R})$ with equal probability. This results in two possible paths through the transformation.

- Given a diffeomorphism $F\colon \mathbb{R}^3 \times S^3 \to \mathbb{R}^3 \times S^3$ we transform $F(\boldsymbol{x}_0, \boldsymbol{q}) = (\boldsymbol{x}_0', \boldsymbol{q}')$.
- Now by using the double cover g, we obtain $\boldsymbol{R}' = g(\boldsymbol{q}')$. It is important to note, that due to the double cover property, we would also obtain $\boldsymbol{R}' = g(-\boldsymbol{q}')$.
- We can then invert the rigid body transform to get $(\boldsymbol{x}_0', \boldsymbol{R}', \boldsymbol{\Psi}) \mapsto \boldsymbol{x}'$ by using $T_{\boldsymbol{\Psi}}^{-1}$

As we explain in the following paragraph, explicit sampling the sign of $\boldsymbol{q}(\boldsymbol{R})$ is not even necessary and you can choose either sign arbitrarily. For completeness, we note that the lift from SO(3) to S^3 could be implemented by any stochastic function, as long as it remains the identity when composed with the projection map. This includes deterministic sign choices as the limiting case. We elaborate on that in appendix B.5.

Volume change induced by the transformation Now let us equip the inputs $\boldsymbol{x}$ with a base density p_0. Furthermore, consider the case where F is invariant under sign flips of $\boldsymbol{q}$ in the first coordinate and equivariant in the second, i.e., we assume that for all $\boldsymbol{x}_0 \in \mathbb{R}^3, \boldsymbol{q} \in S^3$:

$$F(\boldsymbol{x}_0, \boldsymbol{q}) = (\boldsymbol{x}_0', \boldsymbol{q}') \Rightarrow F(\boldsymbol{x}_0, -\boldsymbol{q}) = (\boldsymbol{x}_0', -\boldsymbol{q}'). \tag{8}$$

In this situation, we can compute the total volume change induced by the transformation

$$\boldsymbol{x} \to (\boldsymbol{x}_0, \pm\boldsymbol{q}(\boldsymbol{R})) \to (\boldsymbol{x}_0', \boldsymbol{q}') \to \boldsymbol{x}', \tag{9}$$

independent of the choice of path, as

$$|\boldsymbol{J}_{\boldsymbol{x}\to\boldsymbol{x}'}(\boldsymbol{x})| = |\boldsymbol{J}_F(\boldsymbol{x}_0, \boldsymbol{q}(\boldsymbol{R}))| = |\boldsymbol{J}_F(\boldsymbol{x}_0, -\boldsymbol{q}(\boldsymbol{R}))|\,. \tag{10}$$

This stems from the fact, that the volume contribution of each path is identical and that each path produces exactly the same rotation element at its end. Additionally, the volume change introduced by $T_{\boldsymbol{\Psi}}$ and $T_{\boldsymbol{\Psi}}^{-1}$ cancels out. We give a formal derivation of this transformation law in appendix B.1.

Constructing a flip-symmetric map Given a class of flip-equivariant diffeomorphisms $\Phi(\cdot;\boldsymbol{\theta})\colon S^3 \to S^3$ and arbitrary diffeomorphisms $\xi(\cdot;\boldsymbol{\theta})\colon \mathbb{R}^3 \to \mathbb{R}^3$ we can construct F via the coupling layers (Dinh et al., 2014; 2017):

$$\boldsymbol{x}_0' = \xi(\boldsymbol{x}_0; \boldsymbol{\theta}_\xi(\boldsymbol{q})) \qquad \boldsymbol{q}' = \Phi(\boldsymbol{q}; \boldsymbol{\theta}_\Phi(\boldsymbol{x}_0')). \tag{11}$$

This map is flip-symmetric according to the previous paragraph as long as we assert that $\boldsymbol{\theta}_\xi$ is a flip-invariant conditioning map. We can efficiently invert F and evaluate its volume change according to Eq. (10), whenever ξ and Φ are easy to invert and their change of volume can efficiently be computed. While there exist many candidates that could be chosen for ξ, it is less obvious how to design a suitable family Φ.

3.2. Flip-equivariant diffeomorphisms on S^3

As such, we introduce two classes of smooth and flip-equivariant diffeomorphisms on S^d that can be used to realize F in practice: symmetrized Moebius transforms and projective convex gradient maps. While the first has analytic formulas to compute its volume change and inverse, it is less expressive if one aims to model very multi-modal target densities. As such we consider it as the flip-equivariant S^3 analog of the broadly used real-NVP (Dinh et al., 2017) layers. The second requires more numerical effort to compute its inverse and volume change while being in principle arbitrarily expressive in modeling flip-symmetric multi-modal densities on S^3. We consider it as the flip-equivariant S^3 analog to recently introduced convex-potential flows (Huang et al., 2021).

Symmetrized Moebius transforms A generalized Moebius transform on S^d can be given by

$$\Phi_M(\boldsymbol{p}; \boldsymbol{q}) = \boldsymbol{p} - 2\mathrm{proj}_{\boldsymbol{q}-\boldsymbol{p}}(\boldsymbol{p}) \tag{12}$$

with $\mathrm{proj}_{\boldsymbol{u}}(\boldsymbol{v}) = \frac{\boldsymbol{v}^T\boldsymbol{u}}{\|\boldsymbol{u}\|_2^2}\boldsymbol{v}$. Whenever $\|\boldsymbol{q}\| < 1$ the map $\Phi_M(\cdot;\boldsymbol{q})$ defines a diffeomorphism on S^d (Rezende et al., 2020; Kato & McCullagh, 2020). We can see this using the following intuition: first, send a ray from $\boldsymbol{p}$ through $\boldsymbol{q}$ until it intersects S^3 again at some point $\boldsymbol{p}'$. Then mirror $\boldsymbol{p}'$ onto $-\boldsymbol{p}'$ to get the final result in (12). Each such $\Phi_M(\cdot;\boldsymbol{q})$ defines an involution $\Phi_M(\cdot;\boldsymbol{q}) \circ \Phi_M(\cdot;\boldsymbol{q}) = \mathrm{id}_{S^d}$. Unfortunately, only $\Phi_M(\cdot;\boldsymbol{0}) = \mathrm{id}_{S^d}$ is (trivially) flip-equivariant. However we can use Φ_M to construct the following family of flip-equivariant maps [1]:

$$\Phi_{SM}(\boldsymbol{p}; \boldsymbol{q}) = \frac{\Phi_M(\boldsymbol{p}; \boldsymbol{q}) + \Phi_M(\boldsymbol{p}; -\boldsymbol{q})}{\|\Phi_M(\boldsymbol{p}; \boldsymbol{q}) + \Phi_M(\boldsymbol{p}; -\boldsymbol{q})\|}. \tag{13}$$

If $\|\boldsymbol{q}\| < 1$ each $\Phi_{SM}(\cdot;\boldsymbol{q})$ defines a diffeomorphism on S^d with an analytic inverse. Furthermore, there exists an analytic formula to compute its induced change of volume (see appendix B.3).

Projective convex gradient maps Another construction can be given as follows. Let $\phi\colon \mathbb{R}^{d+1} \to \mathbb{R}$ be a strictly convex and smooth map, with minimizer $\boldsymbol{0}$. Then the normalized gradient map $\Phi\colon S^d \to S^d$, given by

$$\Phi_{CG}(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})}{\|\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\|^2}, \tag{14}$$

[1] See https://www.geogebra.org/m/j7gpwcnf for an interactive animation.

defines a diffeomorphism on S^d (see proof in appendix B.4). Furthermore, if ϕ is flip-invariant, the resulting map Φ_{CG} will be flip-equivariant. While we could model ϕ with arbitrarily complex convex functions, e.g., deep input-convex neural networks (Amos et al., 2017), this requires us to compute the inverse and induced volume change using numeric methods. For S^3 however, we can compute the volume change in closed form efficiently (see appendix B.4).

Relation to Liu et al. (2023) A concurrent approach to modeling flows on $SO(3)$ via the double cover was pursued in Liu et al. (2023). Our work differs significantly in the following aspects:

- While we study physical systems composed of multiple rigid bodies following a joint pose distribution, this prior work studies estimating a single pose in the context of computer vision tasks.
- In order to describe flexible distributions on $SO(3)$ they introduce two flows. The first requires a fiber bundle construction within a specified frame to apply non-smooth spline flows. This construction is fundamentally incompatible with rotational equivariance when modeling multiple poses jointly. The choice of a frame together with the non-smoothness of the induced density makes the approach unsuitable for physical systems as studied in this work.
- The second flow introduced in Liu et al. (2023) is an affine $S^3 \to S^3$ flow, and, as we show in appendix B.4, is a special case of our projective convex gradient maps.
- Interleaving these two flows requires many changes of coordinates between the $SO(3)$ matrices and the S^3 double cover. On the other hand, our Moebius and the projective convex gradient map can model smooth complex multi-modal densities without ever leaving the double-cover construction.

4. Experiments

We show the efficacy of our method for rigid bodies in molecular physics by applying it to sampling two benchmark systems. The first system consists of a single rigid body following a very multi-modal density over the rotations. It serves as a benchmark to see how well the different flow architectures can express multi-modality. The second system consists of an actual molecular crystal in different thermodynamic states.

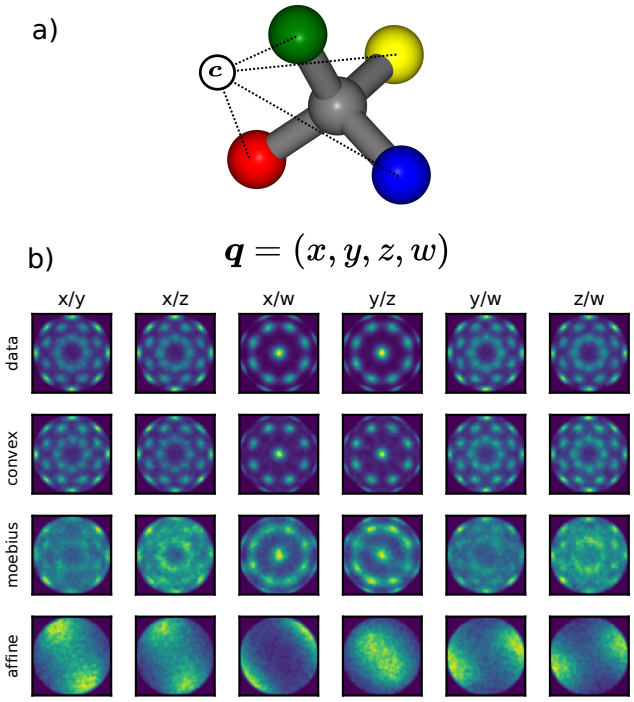

Figure 2. Tetrahedron in an external field: a) each colored bead is attracted with the same force towards the external point $\boldsymbol{c}$ according to the potential defined in Eq. (15). b) Density of rotational degree of freedom. Rotations are represented as unit-quaternions $\boldsymbol{q} = (x, y, z, w)$. First row: density of MD trajectory. Rows 2 to 4: densities of flows using projective convex gradient maps, symmetrized Moebius projections, and affine transformations, respectively.

4.1. Tetrahedron in external field

Our first test system is given by a CH_4 (methane) molecule consisting of 5 atoms (see Fig. 4 a)). We keep internal degrees of freedom constant and fix the carbon to the origin so that only rotational motion is possible. This system interacts with an external field of the form

$$u(\boldsymbol{x}) = C \cdot \sum_{k=1}^{5} \sum_{d=1}^{3} (x_{kd} - c_d)^4, \tag{15}$$

where $\boldsymbol{c} \in \mathbb{R}^3$ and $C > 0$ are control parameters. If $\boldsymbol{c} \neq \boldsymbol{0}$ there are multiple local minima. When sampled in equilibrium, e.g., using an MD simulation, this gives rise to a smooth and multi-modal density on the rotation manifold (see Fig. 2 b) - top row).

As our goal is to model smooth densities on $SO(3)$ we compare the following three models on this system: the affine quaternion flow from (Liu et al., 2023) and the two transforms introduced in Sec. 3.2. Other flow models that could be considered are either not smooth, are not defined on the sub-manifold of interest, or do not possess a scalable

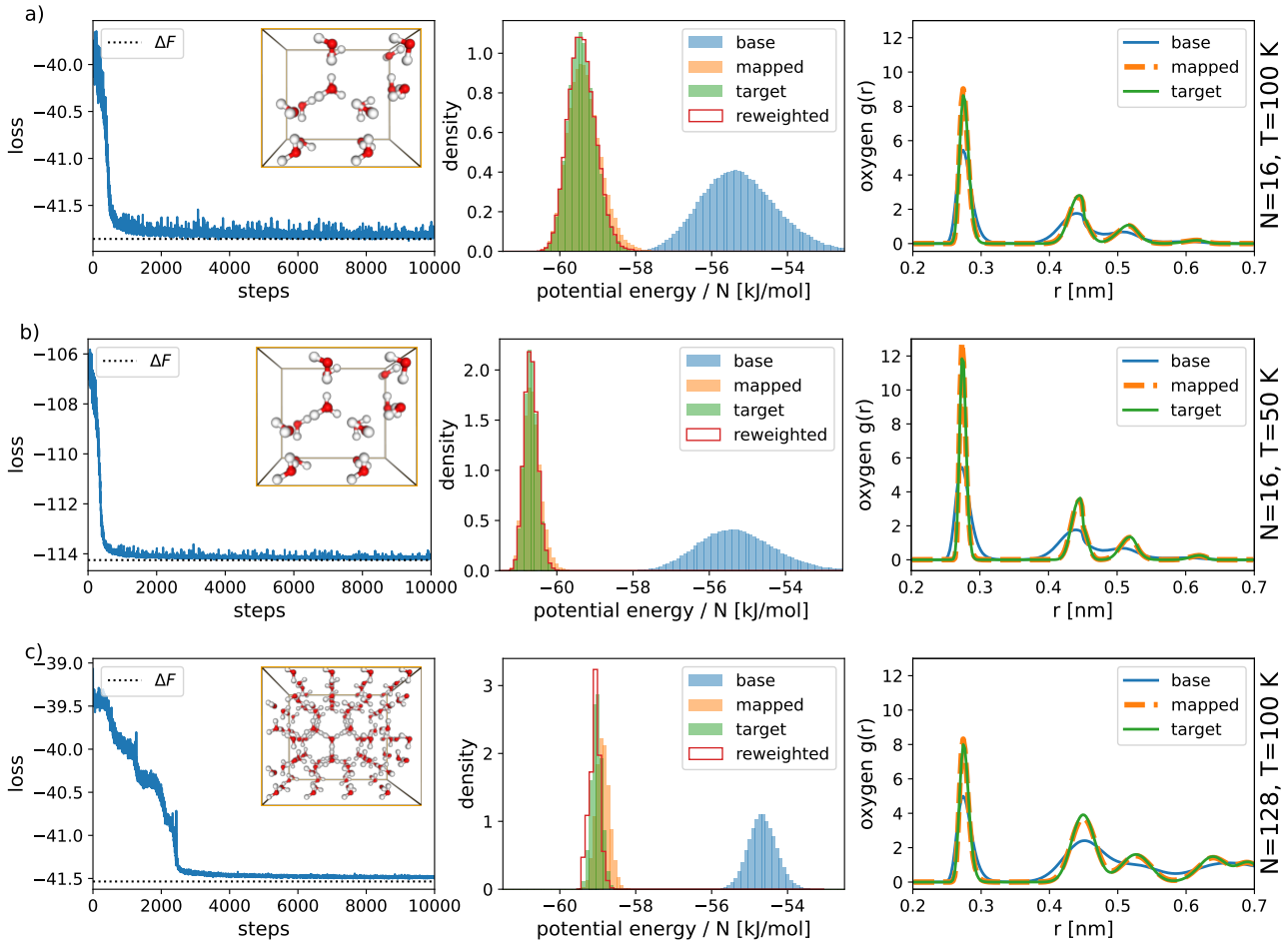

Figure 3. Results for the ice model for different target densities. The temperature of the base density is always T_0 = 250 K, while the temperature T of the target is reported in the figure, as is the number of water molecules N. As expected, the loss approaches from above the reference per-molecule free energy difference ΔF. The estimates reported in Table 1 are instead obtained by applying the LFEP estimator to the evaluation dataset. In all three cases, the map learned by the NF can transform the base density into the target one, as can be seen from the energy distribution and the oxygen-oxygen radial distribution function $g(r)$. The target distribution is not used for training, it is reported merely as a reference.

way to compute exact densities and as such are not suitable for modeling molecular densities. Thus, we do not consider them in this comparison.

We generate an equilibrium dataset of 50,000 samples by sampling from $\mu(\boldsymbol{x}) \propto \exp(-u(\boldsymbol{x}))$ using OpenMM (Eastman et al., 2017) and evaluate the different flow layers by their capability to match the multi-modal structure of its quaternion density when being trained by maximum-likelihood training.

Prior work (Huang et al., 2020; Köhler et al., 2023; Chen et al., 2020; Wu et al., 2020; Nielsen et al., 2020) showed that the expressivity of flows trained on multi-modal datasets can be increased by adding auxiliary Gaussian noise dimensions. We follow this approach and train each of the three tested flow methods on this data set accordingly by minimizing the variational bound to the negative log-likelihood until convergence (see appendix C for details on architecture and training).

Our results, as depicted in Fig. 2, show that projective convex gradient maps can faithfully reconstruct the multi-modal density of the rotations. While the symmetrized Moebius transforms are able to visibly resolve multiple modes, they clearly struggle with the strong multi-modality of the data. The affine quaternion layers can only represent a single mode and as such fail to represent the distribution faithfully. The parameterization of the projective convex potential can be critical for the expressivity of the flow. For brevity, we elaborate on this aspect in appendix C.

4.2. Ice XI in the TIP4P water model

As an example of a molecular crystal, we use water which, while being a simple molecule, is both of primary interest for MD simulations and exhibits highly nontrivial phase behavior (Bore et al., 2022; Kapil et al., 2022). In this experiment, we aim to estimate the free energy difference ΔF between two different thermodynamic conditions, namely a reference temperature T_0 and a lower temperature T. For our simulations we investigate the hydrogen-ordered crystal phase of water, ice XI (Matsumoto et al., 2021), with the TIP4P-Ew rigid water model (Horn et al., 2004). We sample the canonical ensemble, thus fixed number of particles, volume, and temperature. This simple model system does not include quantum mechanical effects and cannot be expected to reproduce experimental measurements (Abascal et al., 2005). However, it is a useful setup to study the Boltzmann distributions of molecular crystals.

The base density μ_0 is given at temperature T_0 = 250 K, thus $u_0(\boldsymbol{x}) = (k_B T_0)^{-1} U(\boldsymbol{x})$, where k_B is the Boltzmann constant and $U(\boldsymbol{x})$ is the TIP4P-Ew force-field energy. We then try to match the density of the same system at a different target temperature T. The quantity ΔF grows when the temperature gap increases, or when increasing the number of particles at a fixed temperature difference. As such, we test our method for the following target potentials:

- For a system composed of $N = 16$ water molecules we estimate ΔF for target temperatures T = 100 K and T = 50 K.
- For a system composed of $N = 128$ water molecules we estimate ΔF for a target temperature T = 100 K.

As reference, we compute the estimate of ΔF from MD simulations using the multistate Bennett acceptance ratio (MBAR) (Shirts & Chodera, 2008) (see appendix A). This method requires an overlap in phase space between the distributions that we want to calculate the free energy difference of. Thus we need to run multiple additional MD simulations at a ladder of intermediate temperatures which can quickly become expensive when ΔF is large (Invernizzi et al., 2022).

By training a NF, we can instead use a single MD run to sample the base, and then use the LFEP estimator to compute ΔF. This can result in a considerable reduction of the computational cost, see appendix D. As proposed by Rizzi et al. (2021) we split the base MD run into two parts, one for training and the other one for the LFEP evaluation, to avoid systematic errors. The used flow consists of coupling layers between positions and rotations according to Fig. 1. We present it in detail in appendix D.

Table 1. Estimates of the free energy difference ΔF per molecule obtained with molecular dynamics (MBAR) and with our normalizing flow (LFEP).

TARGET	MBAR	LFEP
N=16, T=100 K	-41.857 $\pm$ 0.007	-41.859 $\pm$ 0.002
N=16, T=50 K	-114.251 $\pm$ 0.007	-114.252 $\pm$ 0.005
N=128, T=100 K	-41.535 $\pm$ 0.002	-41.534 $\pm$ 0.003

Results The results are shown in Fig. 3. We show that we can achieve a close overlap of the energy distributions between the mapped density from our flow and the target density as obtained by reference MD simulations (second column). We can furthermore reweight the energies and the oxygen-oxygen radial distribution function to achieve nearly perfect overlap (third column).

The ΔF per molecule (thus divided by N) is reported in Table 1. We estimated the error via bootstrapping and report it given as two standard deviations. Other approaches like LBAR (Wirnsberger et al., 2020; 2022) could provide a more accurate estimate but would require samples from the target distribution which we do not assume to be available in our experiments.

5. Discussion

In this work, we presented a new approach to approximate the densities of multiple interacting molecules by modeling their positions and orientations using normalizing flows. A key element of this was a derivation of a smooth flow structure using the quaternion double cover and providing an efficient implementation via two categories of flip-equivariant flows on S^3. We furthermore demonstrated the effectiveness of this approach for modeling densities of molecular crystals by evaluating it on a multi-modal benchmark system and a range of ice systems.

We note that beyond the very important application to molecular crystals, rigid body flows could also become relevant in other domains, such as robotics, evidenced by related work like Brehmer et al. (2023).

Limitations and possible extensions While the result for ice XI is promising and paves the way for many interesting applications of normalizing flows in the field of molecular crystals, a major challenge ahead is dealing with phase transitions or even going beyond the crystal phase to liquid and gas. However, these are still open problems even in the case of non-molecular systems, e.g., when monatomic crystals are modeled Wirnsberger et al. (2022). An interesting but nontrivial next step would be extending the present architecture with a flow model for the positions that can handle

fluids and phase transitions.

A second aspect that we did not explore further in this work is exploiting the $SE(3)$ symmetry of jointly moving all rigid bodies. Both introduced flow layers can easily be extended to fully rotation equivariant architectures by making the learnable functions $\xi, \theta_\xi, \theta_\Phi$ in Eq. (11) equivariant, with architectures such as EGNN (Garcia Satorras et al., 2021), NequIP (Batzner et al., 2022) or MACE (Batatia et al., 2022). Such architectures can also compute pairwise interactions equivariant to jointly moving pairs of rigid bodies. Furthermore, many rigid bodies have internal symmetries, such as the mirror symmetry of the water molecule. For N water molecules, this gives a symmetry group of order 2^N. To scale to larger systems, built-in equivariance to this group may be necessary.

It is important to note that while here we only consider rigid-body molecules, the proposed flow architecture can be straightforwardly extended to incorporate the internal degrees of freedom of the molecules. This is an important aspect, as it is essential to handle larger molecules or more accurate force fields.

Finally, recent work of Abbott et al. (2022) raised questions about the scaling limits of normalizing flows when sampling physical potentials in lattice physics. Although such a study has not yet been carried out for molecular systems, it will be important to understand how this result relates to the sampling of molecular crystals and whether flow-based approaches can be reliably and efficiently scaled to much larger systems.

Software and Data

All the code used to obtain the results is available at `https://github.com/noegroup/rigid-flows`.

Acknowledgements

We thank Andreas Krämer for his invaluable editorial support in preparing this version of the manuscript and for his insightful advice. We furthermore thank Maaike Galama for helpful discussions about MBAR.

J.K and F.N. acknowledge funding by DFG CRC1114 Project B08, DFG RTG DAEDALUS, ERC consolidator grant 772230. M.I. acknowledges support from the Humboldt Foundation for a Postdoctoral Research Fellowship.

References

Abascal, J. L. F., Sanz, E., García Fernández, R., and Vega, C. A potential model for the study of ices and amorphous water: TIP4P/Ice. *The Journal of Chemical Physics*, 122 (23):234511, 2005.

Abbott, R., Albergo, M. S., Botev, A., Boyda, D., Cranmer, K., Hackett, D. C., Matthews, A. G. D. G., Racanière, S., Razavi, A., Rezende, D. J., Romero-López, F., Shanahan, P. E., and Urban, J. M. Aspects of scaling and scalability for flow-based sampling of lattice QCD. *arXiv preprint arXiv:2211.07541*, 2022.

Ahmad, R. and Cai, W. Free energy calculation of crystalline solids using normalizing flows. *Modelling and Simulation in Materials Science and Engineering*, 30(6): 065007, 2022.

Albergo, M., Kanwar, G., and Shanahan, P. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155. PMLR, 2017.

Batatia, I., Kovacs, D. P., Simm, G. N. C., Ortner, C., and Csanyi, G. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, 2022.

Ben-Hamu, H., Cohen, S., Bose, J., Amos, B., Nickel, M., Grover, A., Chen, R. T. Q., and Lipman, Y. Matching normalizing flows and probability paths on manifolds. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1749–1763. PMLR, 2022.

Bernstein, J. *Polymorphism in Molecular Crystals*. Oxford University Press, 2020.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022.

Bore, S. L., Piaggi, P. M., Car, R., and Paesani, F. Phase diagram of the TIP4P/Ice water model by enhanced sampling simulations. *The Journal of Chemical Physics*, 157 (5):054504, 2022.

Boyda, D., Kanwar, G., Racanière, S., Rezende, D. J., Albergo, M. S., Cranmer, K., Hackett, D. C., and Shanahan, P. E. Sampling using $\mathrm{SU}(n)$ gauge equivariant flows. *Phys. Rev. D*, 103:074504, 2021.

Brehmer, J., Bose, J., Haan, P. D., and Cohen, T. EDGI: Equivariant diffusion for planning with embodied agents. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

Chen, J., Lu, C., Chenli, B., Zhu, J., and Tian, T. Vflow: More expressive generative flows with variational data augmentation. In *International Conference on Machine Learning*, pp. 1660–1669. PMLR, 2020.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Advances in neural information processing systems*, pp. 6571–6583, 2018.

Cohen, S., Amos, B., and Lipman, Y. Riemannian convex potential maps. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2028–2038. PMLR, 2021.

Coretti, A., Falkner, S., Geissler, P., and Dellago, C. Learning Mappings between Equilibrium States of Liquid Systems Using Normalizing Flows. *arXiv preprint arXiv:2208.10420*, 2022.

Dibak, M., Klein, L., Krämer, A., and Noé, F. Temperature Steerable Flows and Boltzmann Generators. *Physical Review Research*, 4(4):L042005, 2021.

Ding, X. and Zhang, B. DeepBAR: A Fast and Exact Method for Binding Free Energy Computation. *The Journal of Physical Chemistry Letters*, 12:2509–2515, 2021.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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., et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7): e1005659, 2017.

Falorsi, L. Continuous normalizing flows on manifolds. *arXiv preprint arXiv:2104.14959*, 2021.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3244–3253. PMLR, 2019.

Frenkel, D. and Ladd, A. J. C. New Monte Carlo method to compute the free energy of arbitrary solids. Application to the fcc and hcp phases of hard spheres. *The Journal of Chemical Physics*, 81(7):3188–3193, 1984.

Frenkel, D. and Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, New York, 2001.

Gabrié, M., Rotskoff, G. M., and Vanden-Eijnden, E. Adaptive Monte Carlo augmented with normalizing flows. *Proc. Natl. Acad. Sci. U.S.A.*, 119(10):e2109420119, 2022.

Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34: 4181–4192, 2021.

Gemici, M. C., Rezende, D., and Mohamed, S. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Horn, H. W., Swope, W. C., Pitera, J. W., Madura, J. D., Dick, T. J., Hura, G. L., and Head-Gordon, T. Development of an improved four-site water model for biomolecular simulations: TIP4P-Ew. *The Journal of Chemical Physics*, 120(20):9665–9678, 2004.

Huang, C.-W., Dinh, L., and Courville, A. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

Huang, C.-W., Chen, R. T. Q., Tsirigotis, C., and Courville, A. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021.

Invernizzi, M., Krämer, A., Clementi, C., and Noé, F. Skipping the replica exchange ladder with normalizing flows. *The Journal of Physical Chemistry Letters*, 13(50):11643–11649, 2022. PMID: 36484770.

Jarzynski, C. Targeted free energy perturbation. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 65(4):5, 2002.

Kalatzis, D., Ye, J. Z., Pouplin, A., Wohlert, J., and Hauberg, S. Density estimation on smooth manifolds with normalizing flows. *arXiv preprint arXiv:2106.03500*, 2021.

Kapil, V., Schran, C., Zen, A., Chen, J., Pickard, C. J., and Michaelides, A. The first-principles phase diagram of monolayer nanoconfined water. *Nature*, 609(7927): 512–516, 2022.

Kato, S. and McCullagh, P. Some properties of a cauchy family on the sphere derived from the möbius transformations. *Bernoulli*, 26(4), 2020.

Katsman, I., Lou, A., Lim, D., Jiang, Q., Lim, S. N., and De Sa, C. M. Equivariant manifold flows. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10600–10612. Curran Associates, Inc., 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Kish, L. Sampling Organizations and Groups of Unequal Sizes. *American Sociological Review*, 30(4):564, 1965.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pp. 5361–5370. PMLR, 2020.

Köhler, J., Krämer, A., and Noé, F. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021.

Köhler, J., Chen, Y., Krämer, A., Clementi, C., and Noé, F. Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 2023. PMID: 36668906.

Li, S.-H. and Wang, L. Neural network renormalization group. *Physical review letters*, 121(26):260601, 2018.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Liu, Y., Liu, H., Yin, Y., Wang, Y., Chen, B., and Wang, H. Delving into Discrete Normalizing Flows on SO(3) Manifold for Probabilistic Rotation Modeling, 2023.

Lou, A., Lim, D., Katsman, I., Huang, L., Jiang, Q., Lim, S. N., and De Sa, C. M. Neural manifold ordinary differential equations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17548–17558. Curran Associates, Inc., 2020.

Mathieu, E. and Nickel, M. Riemannian continuous normalizing flows. *Advances in Neural Information Processing Systems*, 33, 2020.

Matsumoto, M., Yagasaki, T., and Tanaka, H. Novel Algorithm to Generate Hydrogen-Disordered Ice Structures. *Journal of Chemical Information and Modeling*, 61(6): 2542–2546, 2021.

Midgley, L. I., Stimper, V., Simm, G. N. C., Schölkopf, B., and Hernández-Lobato, J. M. Flow annealed importance sampling bootstrap, 2022.

Müller, T., Mcwilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural Importance Sampling. *ACM Transactions on Graphics*, 38(5):1–19, 2019.

Murphy, K. A., Esteves, C., Jampani, V., Ramalingam, S., and Makadia, A. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7882–7893. PMLR, 2021.

Nicoli, K. A., Nakajima, S., Strodthoff, N., Samek, W., Müller, K.-R., and Kessel, P. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2):023304, 2020.

Nicoli, K. A., Anders, C. J., Funcke, L., Hartung, T., Jansen, K., Kessel, P., Nakajima, S., and Stornati, P. Estimation of thermodynamic observables in lattice field theories with deep generative models. *Physical review letters*, 126 (3):032001, 2021.

Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., and Welling, M. Survae flows: Surjections to bridge the gap between vaes and flows. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015.

Rezende, D. J. and Racanière, S. Implicit riemannian concave potential maps. *arXiv preprint arXiv:2110.01288*, 2021.

Rezende, D. J., Papamakarios, G., Racaniere, S., Albergo, M., Kanwar, G., Shanahan, P., and Cranmer, K. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, pp. 8083–8092. PMLR, 2020.

Rizzi, A., Carloni, P., and Parrinello, M. Targeted Free Energy Perturbation Revisited: Accurate Free Energies from Mapped Reference Potentials. *The Journal of Physical Chemistry Letters*, 12(39):9449–9454, 2021.

Sbailò, L., Dibak, M., and Noé, F. Neural mode jump monte carlo. *The Journal of Chemical Physics*, 154(7):074101, 2021.

Schieber, N. P. and Shirts, M. R. Configurational mapping significantly increases the efficiency of solid-solid phase coexistence calculations via molecular dynamics: Determining the FCC-HCP coexistence line of Lennard-Jones particles. *The Journal of Chemical Physics*, 150(16): 164112, 2019.

Schieber, N. P., Dybeck, E. C., and Shirts, M. R. Using reweighting and free energy surface interpolation to predict solid-solid phase diagrams. *The Journal of Chemical Physics*, 148(14):144104, 2018.

Shirts, M. R. and Chodera, J. D. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of Chemical Physics*, 129(12):124105, 2008.

Tabak, E. G., Vanden-Eijnden, E., et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vega, C., Sanz, E., Abascal, J. L. F., and Noya, E. G. Determination of phase diagrams via computer simulation: Methodology and applications to water, electrolytes and proteins. *Journal of Physics: Condensed Matter*, 20(15): 153101, 2008.

Wirnsberger, P., Ballard, A., Papamakarios, G., Abercrombie, S., Racanière, S., Pritzel, A., Blundell, C., et al. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112–144112, 2020.

Wirnsberger, P., Papamakarios, G., Ibarz, B., Racanière, S., Ballard, A. J., Pritzel, A., and Blundell, C. Normalizing flows for atomic solids. *Machine Learning: Science and Technology*, 3(2):025009, 2022.

Wu, H., Köhler, J., and Noe, F. Stochastic normalizing flows. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5933–5944. Curran Associates, Inc., 2020.

Xu, M., Luo, S., Bengio, Y., Peng, J., and Tang, J. Learning neural generative dynamics for molecular conformation generation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Zhang, Z., Liu, X., Yan, K., Tuckerman, M. E., and Liu, J. Unified Efficient Thermostat Scheme for the Canonical Ensemble with Holonomic or Isokinetic Constraints via Molecular Dynamics. *The Journal of Physical Chemistry A*, 123(28):6056–6079, 2019.

Zwanzig, R. W. High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

A. The sampling problem in molecular crystals

Molecular crystals are of great interest for several important applications, but there are many open problems when it comes to efficiently characterize their properties via computer simulations. One of the reasons is that there are an exponentially high number of energetically stable polymorphs, but at any given thermodynamic condition only few are stable enough to be observed experimentally, and only one is the most stable one. Even for a simple molecule like water, more than 20 crystal polymorphs have been observed [2]. This poses several challenges, that are typically tackled with different methodologies. Here we do not focus on how to find all the energetically stable polymorphs, but rather on the stability at non-zero temperature, where entropic effects are important and thus free energy differences must be estimated, instead of just energy differences.

One of the most popular ways of computing free energy differences for atomic and molecular crystals, is thermodynamic integration (TI) (Frenkel & Smit, 2001). As an example, let us consider two states, A and B, with energies $u_A(\mathbf{x})$ and $u_B(\mathbf{x})$. To estimate the free energy difference ΔF_{AB} with TI, one has to run multiple MD or MCMC simulations of the system along an interpolation between u_A and u_B, such that each simulation samples a region of the phase space $\mathbf{x}$ that has some overlap with the closest ones. In the example considered in our paper, A and B are simply two different temperatures and the interpolation is done by slowly changing the temperature, but one can also perform TI between a physical state and a reference ideal normal distribution, usually referred as Einstein crystal in the literature (Frenkel & Smit, 2001). Once these simulations have been performed, the actual estimate of ΔF_{AB} can be obtained with various postprocessing methods, but a typical choice is the MBAR method, that has been proved to provide the lowest variance estimator (Shirts & Chodera, 2008).

Possibly, the most straightforward way of estimating ΔF_{AB} is to use the free energy perturbation formula (Zwanzig, 1954):

$$\Delta F_{AB} = -\log\langle e^{u_A - u_B}\rangle_A = \log\langle e^{u_B - u_A}\rangle_B \tag{16}$$

which requires samples either from A or B. It is important to notice that a good overlap in configuration space is crucial, otherwise the variance of the ensemble average is orders of magnitude larger than ΔF_{AB}. Sampling a ladder of overlapping intermediate states allows one to use this formula to estimate ΔF_{AB} one step at the time. The MBAR method is based on the same idea, but uses a self consistent procedure to combine all the samples and obtain an estimator which minimizes the variance (Shirts & Chodera, 2008).

The main drawback of TI is that it can be computationally extremely expensive, requiring sampling from several intermediate states that are of no direct interest. This is especially exacerbated in the case of molecular crystals, where the integration path can be highly nontrivial and where to obtain accurate potential energies one must perform expensive quantum mechanical calculations. To avoid this expensive calculation, Jarzynski (2002) proposed to use an explicit invertible map to bridge A and B, the so called targeted free energy perturbation method. Defining such maps is far from trivial, even for the simplest systems, that is why the method has rarely been used. However, recently Wirnsberger et al. (2020) proposed to use normalizing flows to learn such maps, which gave rise to the LFEP method that is used also in this work.

B. Proofs and derivations

B.1. Volume change on manifolds

We follow the notation of Rezende et al. (2020) Appendix A. Let $\mathcal{M}$ and $\mathcal{N}$ be m and n dimensional submanifolds of $\mathbb{R}^d$ respectively and $F\colon \mathcal{M} \to \mathcal{N}$ a smooth injective map that can be extended to open neighborhoods of $\mathcal{M}$ and $\mathcal{N}$. Then we can compute its induced change of volume as follows: let $T_{\boldsymbol{x}}\mathcal{M}$ and $T_{\boldsymbol{F}(\boldsymbol{x})}\mathcal{N}$ be the tangent spaces at a point $\boldsymbol{x} \in \mathcal{M}$ and its image $F(\boldsymbol{x}) \in \mathcal{N}$. Furthermore, let $\boldsymbol{E}_{\boldsymbol{x}} \in \mathbb{R}^{d\times n}$ and $\boldsymbol{E}_{F(\boldsymbol{x})} \in \mathbb{R}^{d\times m}$ be bases of $T_{\boldsymbol{x}}\mathcal{M}$ and $T_{\boldsymbol{F}(\boldsymbol{x})}\mathcal{N}$ respectively. Then

$$|\boldsymbol{J}_F(\boldsymbol{x})| = \sqrt{\det \boldsymbol{E}_{\boldsymbol{x}}^T \boldsymbol{J}_F^T(\boldsymbol{x}) \boldsymbol{J}_F(\boldsymbol{x}) \boldsymbol{E}_{\boldsymbol{x}}}. \tag{17}$$

If $m = n$ we also have

$$|\boldsymbol{J}_F(\boldsymbol{x})| = \det \boldsymbol{E}_{F(\boldsymbol{x})}^T \boldsymbol{J}_F(\boldsymbol{x}) \boldsymbol{E}_{\boldsymbol{x}}. \tag{18}$$

[2] https://en.wikipedia.org/wiki/Ice

B.2. Density of the mixture

Let p_0 be a density over the inputs $\boldsymbol{x}$. Furthermore, define

$$A_+(\boldsymbol{x}_0, \boldsymbol{R}) = (\boldsymbol{x}_0, \boldsymbol{q}(\boldsymbol{R})),\ A_-(\boldsymbol{x}_0, \boldsymbol{R}) = (\boldsymbol{x}_0, -q(\boldsymbol{R})), \tag{19}$$

Summing up the probabilities of each path and accounting for the induced volume change of each transformation, we obtain the mixture density

$$\begin{aligned} p(\boldsymbol{x}') =& \tfrac{1}{2} \cdot |\boldsymbol{J}_{T_{\boldsymbol{\Psi}}}(\boldsymbol{x}')| \cdot |\boldsymbol{J}_{F^{-1}}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot |\boldsymbol{J}_{T_{\boldsymbol{\Psi}}^{-1}}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) \\ &+ \tfrac{1}{2} \cdot |\boldsymbol{J}_{T_{\boldsymbol{\Psi}}}(\boldsymbol{x}')| \cdot |\boldsymbol{J}_{F^{-1}}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot |\boldsymbol{J}_{T_{\boldsymbol{\Psi}}}^{-1}(F^{-1}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) \end{aligned} \tag{20}$$

First, we see the following: after $T_{\boldsymbol{\Psi}}$ maps $\boldsymbol{\Psi}$ into the standard frame any change in $\boldsymbol{x}_0$ or $\boldsymbol{R}$ is merely a SE(3) action and as such does not contribute to the volume. From that, we get that $|\boldsymbol{J}_{T_{\boldsymbol{\Psi}}}(\boldsymbol{x}')| \cdot |\boldsymbol{J}_{T_{\boldsymbol{\Psi}}^{-1}}(F^{-1}(A_\pm(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| = 1$.

Now let F be flip-symmetric according to the definition in Sec. 3.2.

From the definition of F and using the double cover we get $T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))) = T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}'))))$.

We furthermore get

$$\boldsymbol{J}_F(\boldsymbol{x}, -\boldsymbol{q}) = \underbrace{\begin{bmatrix} \boldsymbol{I}_{3\times 3} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_{4\times 4} \end{bmatrix}}_{\boldsymbol{B}:=} \boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q}). \tag{21}$$

Let $\boldsymbol{E}$ be a basis according to Sec. B.1. Then we immediately see that

$$\begin{aligned} |\boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q})| &= \sqrt{\det \boldsymbol{E}^T \boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q})^T \boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q}) \boldsymbol{E}} && (22) \\ &= \sqrt{\det \boldsymbol{E}^T \boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q})^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{J}_F(\boldsymbol{x}, \boldsymbol{q}) \boldsymbol{E}} && (23) \\ &= \sqrt{\det \boldsymbol{E}^T \boldsymbol{J}_F(\boldsymbol{x}, -\boldsymbol{q})^T \boldsymbol{J}_F(\boldsymbol{x}, -\boldsymbol{q}) \boldsymbol{E}} && (24) \\ &= |\boldsymbol{J}_F(\boldsymbol{x}, -\boldsymbol{q})| && (25) \end{aligned}$$

Combining this we can simplify Eq. (20) into

$$\begin{aligned} p(\boldsymbol{x}') =& \tfrac{1}{2} \cdot |\boldsymbol{J}_{F^{-1}}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) \\ &+ \tfrac{1}{2} \cdot |\boldsymbol{J}_{F^{-1}}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_-(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) && (26) \\ =& \tfrac{1}{2} \cdot |\boldsymbol{J}_{F^{-1}}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) \\ &+ \tfrac{1}{2} \cdot |\boldsymbol{J}_{F^{-1}}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) && (27) \\ =& |\boldsymbol{J}_{F^{-1}}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))| \cdot p_0(T_{\boldsymbol{\Psi}}^{-1}(F^{-1}(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}')))))) && (28) \end{aligned}$$

by substituting $\boldsymbol{x}' = \boldsymbol{x}'(\boldsymbol{x})$ and using the short-hand notation $|\boldsymbol{J}_F(\boldsymbol{x}_0, \boldsymbol{q}(\boldsymbol{R}))| = |\boldsymbol{J}_F(A_+(T_{\boldsymbol{\Psi}}(\boldsymbol{x}'))|$ we end up with the formula for the induced volume change:

$$p_0(\boldsymbol{x}) = p(\boldsymbol{x}'(\boldsymbol{x}))\, |\boldsymbol{J}_F(\boldsymbol{x}_0, \boldsymbol{q}(\boldsymbol{R}))|\,. \tag{29}$$

B.3. Derivations for symmetrized Moebius transform

A generalized Moebius transform $S^d \to S^d$, given a parameter $\boldsymbol{q} \in B^d = \{\boldsymbol{q} \in \mathbb{R}^{d+1} \mid |\boldsymbol{q}| < 1\}$ can be given by

$$\Phi_M(\boldsymbol{p}; \boldsymbol{q}) = \boldsymbol{p} - 2\mathrm{proj}_{\boldsymbol{q}-\boldsymbol{p}}(\boldsymbol{p}) \tag{30}$$

with $\mathrm{proj}_{\boldsymbol{u}}(\boldsymbol{v}) = \frac{\boldsymbol{v}^T\boldsymbol{u}}{\|\boldsymbol{u}\|_2^2}\boldsymbol{v}$. This map is an involutive diffeomorphism with $\Phi_M(\Phi_M(\boldsymbol{p};\boldsymbol{q});\boldsymbol{q}) = \boldsymbol{p}$. The sign-symmetrized variant is

$$\Phi_{SM}(\boldsymbol{p};\boldsymbol{q}) = \frac{\Phi_M(\boldsymbol{p};\boldsymbol{q}) + \Phi_M(\boldsymbol{p};-\boldsymbol{q})}{\|\Phi_M(\boldsymbol{p};\boldsymbol{q}) + \Phi_M(\boldsymbol{p};-\boldsymbol{q})\|}. \tag{31}$$

which satisfies $\Phi_{SM}(\boldsymbol{p};\boldsymbol{q}) = \Phi_{SM}(\boldsymbol{p};-\boldsymbol{q})$.

Both maps are equivariant to the choice of orthonormal coordinates, as for any coordinate transformation $g \in O(d)$, $\Phi_M(g\boldsymbol{p};g\boldsymbol{q}) = g\Phi_M(\boldsymbol{p};\boldsymbol{q})$ and then by linearity $\Phi_{SM}(g\boldsymbol{p};g\boldsymbol{q}) = g\Phi_{SM}(\boldsymbol{p};\boldsymbol{q})$. Combined with the $\boldsymbol{q} \mapsto -\boldsymbol{q}$ invariance of Φ_{SM}, this implies to the desired sign-equivariance: $\Phi_{SM}(-\boldsymbol{p};\boldsymbol{q}) = -\Phi_{SM}(\boldsymbol{p};\boldsymbol{q})$.

Due to the $O(d)$ equivariance, we're free to analyze the maps in a convenient coordinate system. Without loss of generality, we can place point $\boldsymbol{p} \in S^d$ at $(x, \sqrt{1-x^2}, 0, 0, ...)$ and $\boldsymbol{q} \in B^d$ at $(r, 0, 0, ...)$. It is easy to see that the image $\Psi_{SM}(\boldsymbol{p};\boldsymbol{q}) = (x', y', 0, 0, ...)$ has zeros in all but the first two coordinates. Also, the coordinates are given by the planar $d = 1$ version of the transformation: $(x', y') = \Phi_{SM}((x, \sqrt{1-x^2}), (r, 0))$.

Using a computer algebra system, we can simplify this expression to find:

$$x' = \frac{x(r^2-1)}{\sqrt{1+r^4+r^2(2-4x)}}$$
$$y' = -\sqrt{1-x'^2}$$

Invertibility Given r, the map $x \mapsto x'$ can be inverted via a computer algebra system to find:

$$x = \frac{-x'(r^2+1)}{\sqrt{1+r^4+r^2(4x'^2-2)}}$$

By assumption, the y coordinate was in the positive half-plane, so $y = \sqrt{1-x^2}$.

To compute the general inverse $\boldsymbol{p} = \Phi_{SM}^{-1}(\boldsymbol{p}';\boldsymbol{q})$, we compute a $g \in O(d)$ so that $g\boldsymbol{q} = (r, 0, ...)$ and $g\boldsymbol{p}' = (x', -\sqrt{1-x'^2}, ...)$. Then we use the above inverse to find $g\boldsymbol{p} = (x, \sqrt{1-x^2}, 0, ...)$ and find $\boldsymbol{p} = g^{-1}(x, \sqrt{1-x^2}, 0, ...)$.

Change of volume For the change of volume of the symmetric Moebius transformation, we focus on the case $d = 3$ of the three-sphere. Now, a convenient parametrization (without loss of generality) is $\boldsymbol{p} = (1, 0, 0, 0)$ and $\boldsymbol{q} = (\sqrt{r^2 - q_y^2}, q_y, 0, 0)$. We'll omit the argument $\boldsymbol{q}$ in $\boldsymbol{p}' = \Phi_{SM}(\boldsymbol{p};\boldsymbol{q})$ going forward.

Then, using the embedding $\iota : S^3 \hookrightarrow \mathbb{R}^4$, we can embed the tangent space in the ambient space $d\iota_{\boldsymbol{p}} : T_{\boldsymbol{p}}S^3 \hookrightarrow \mathbb{R}^4$, where it is given by the vectors $(0, v_1, v_2, v_3)$ for $v \in \mathbb{R}^3$.

In the tangent direction $v \in T_{\boldsymbol{p}}S^3$, the direction of change of $\boldsymbol{p}'$ is given by $\frac{\partial \Phi_{SM}(\boldsymbol{p}+tv)}{\partial t}|_{t=0} \in \mathbb{R}^4$. Thus, the Jacobian matrix, in standard coordinates of the tangent plane, is given by:

$$J(\boldsymbol{p})E_{\boldsymbol{p}} = \left(\frac{\partial \Phi_{SM}((1,t,0,0))}{\partial t}|_{t=0}, \frac{\partial \Phi_{SM}((1,0,t,0))}{\partial t}|_{t=0}, \frac{\partial \Phi_{SM}((1,0,0,t))}{\partial t}|_{t=0}\right) \in \mathbb{R}^{4\times 3}$$

Via a computer algebra system, we can compute and simplify the change of volume to get:

$$\sqrt{\det(E_{\boldsymbol{p}}^T J(\boldsymbol{p})^T J(\boldsymbol{p}) E_{\boldsymbol{p}})} = \frac{(1-r^2)(1+r^2)^3}{(4q_y^2 + (1-r^2)^2)^2}$$

In the general case, the change of volume is also given by the above formula with $r = |\boldsymbol{q}|$ and $q_y^2 = r^2 - \langle \boldsymbol{p}, \boldsymbol{q}\rangle^2$.

B.4. Derivations for projective convex gradient maps

Invertibility Following Sec. 3.2 we assume that $\phi\colon \mathbb{R}^{d+1} \to \mathbb{R}$ is a strictly convex function with minimum at $\mathbf{0}$.

Theorem B.1. *The projective convex gradient map $\Phi(\boldsymbol{p})\colon S^d \to S^d, \boldsymbol{p} \mapsto \frac{\nabla_{\boldsymbol{p}}\phi(\boldsymbol{p})}{\|\nabla_{\boldsymbol{p}}\phi(\boldsymbol{p})\|}$ is a diffeomorphism.*

Proof. Let $\boldsymbol{p} \in S^d$ and $\boldsymbol{p}' = \Phi(\boldsymbol{p})$ its image under the projective convex gradient map. Let furthermore $T_{\boldsymbol{p}}, T_{\boldsymbol{p}'}$ be the tangent spaces at $\boldsymbol{p}, \boldsymbol{p}'$, respectively. Furthermore let $\boldsymbol{E}_{\boldsymbol{p}}, \boldsymbol{E}_{\boldsymbol{p}'} \in \mathbb{R}^{(d+1)\times d}$ be ortho-normal bases of the two tangent spaces. Then it suffices to show that $\boldsymbol{A} := \boldsymbol{E}_{\boldsymbol{p}'}^T \boldsymbol{J}_\Phi(\boldsymbol{p}) \boldsymbol{E}_{\boldsymbol{p}} \in \mathbb{R}^{d\times d}$ is a non-singular matrix with a non-zero determinant. Denote $\boldsymbol{g}(\boldsymbol{p}) := \nabla_{\boldsymbol{p}} \phi(\boldsymbol{p})$. By using the chain rule, we first see that

$$\boldsymbol{J}_{\Phi_{CG}}(\boldsymbol{p}) = \frac{\mathcal{H}_\phi(\boldsymbol{p})}{\|\boldsymbol{g}(\boldsymbol{p})\|} - \frac{\boldsymbol{g}(\boldsymbol{p})\boldsymbol{g}(\boldsymbol{p})^T}{\|\boldsymbol{g}(\boldsymbol{p})\|^3} \mathcal{H}_\phi(\boldsymbol{p}) \tag{32}$$

$$= \left(\boldsymbol{I} - \boldsymbol{p}'\boldsymbol{p}'^T\right) \frac{\mathcal{H}_\phi(\boldsymbol{p})}{\|\boldsymbol{g}(\boldsymbol{p})\|} \tag{33}$$

We prove by contradiction: Assume $\boldsymbol{A}$ is singular and that $\boldsymbol{v}$ is a unit vector with $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{0}$ and denote $\boldsymbol{w} = \boldsymbol{E}_{\boldsymbol{p}}^T \boldsymbol{v}$. Since $\boldsymbol{E}_{\boldsymbol{p}}$ is a basis of $T_{\boldsymbol{p}}$ we have $\boldsymbol{w} \neq \boldsymbol{0}$ and furthermore $\boldsymbol{w} \in T_{\boldsymbol{p}}$.

Now because $\boldsymbol{E}_{\boldsymbol{p}}$ is full rank on its image we have $\boldsymbol{J}_\Phi(\boldsymbol{p})\boldsymbol{w} = \boldsymbol{0}$.

Since ϕ is strictly convex $\mathcal{H}_\phi(\boldsymbol{p})$ is a strictly positive definite matrix and thus we know that $\mathcal{H}_\phi(\boldsymbol{p})\boldsymbol{w} \neq \boldsymbol{0}$. Thus we have

$$\boldsymbol{0} = \left(\boldsymbol{I} - \boldsymbol{p}'\boldsymbol{p}'^T\right) \frac{\mathcal{H}_\phi(\boldsymbol{p})}{\|\boldsymbol{g}(\boldsymbol{p})\|} \boldsymbol{w} \quad \Leftrightarrow \quad \mathcal{H}_\phi(\boldsymbol{p})\boldsymbol{w} = \boldsymbol{p}'\boldsymbol{p}'^T \mathcal{H}_\phi(\boldsymbol{p})\boldsymbol{w} \tag{34}$$

And as such $\mathcal{H}_\phi(\boldsymbol{p})\boldsymbol{w} \propto \boldsymbol{p}' \propto \boldsymbol{g}(\boldsymbol{p}) \implies \boldsymbol{w} \propto \mathcal{H}_\phi^{-1}(\boldsymbol{p})\boldsymbol{g}(\boldsymbol{p})$. Thus, $\mathcal{H}_\phi^{-1}(\boldsymbol{p})\boldsymbol{g}(\boldsymbol{p}) \in T_{\boldsymbol{p}} \implies \left(\mathcal{H}_\phi^{-1}(\boldsymbol{p})\boldsymbol{g}(\boldsymbol{p})\right)^T \boldsymbol{p} = 0$.

Now set $\boldsymbol{G} = \mathcal{H}_\phi^{-1}(\boldsymbol{p})$ keeping $\boldsymbol{p}$ fixed and define the function $\psi(\boldsymbol{q}) = \phi(\boldsymbol{G}\boldsymbol{q})$. As $\boldsymbol{G}$ is strictly positive and ϕ is strictly convex with minimum $\phi(\boldsymbol{0})$ this new function ψ is strictly convex with minimum $\psi(\boldsymbol{0})$ as well.

Now we can use the strict convexity of ψ to get

$$\psi(\boldsymbol{0}) > \psi(\boldsymbol{p}) + \nabla_{\boldsymbol{p}}\psi(\boldsymbol{p})^T(\boldsymbol{0} - \boldsymbol{p}) \tag{35}$$

$$= \psi(\boldsymbol{p}) - (\boldsymbol{G}\nabla_{\boldsymbol{p}}\phi(\boldsymbol{p}))^T \boldsymbol{p} \tag{36}$$

$$= \psi(\boldsymbol{p}) - \left(\mathcal{H}_\phi^{-1}(\boldsymbol{p})\boldsymbol{g}(\boldsymbol{p})\right)^T \boldsymbol{p} \tag{37}$$

$$= \psi(\boldsymbol{p}) \tag{38}$$

However, this contradicts $\boldsymbol{0}$ being the minimum of ψ. □

Parameterizing the potential ϕ While ϕ could be modeled by any general input convex neural network (Amos et al., 2017) with minimizer $\boldsymbol{0}$ we decided on a very simple implementation that worked well in practice and is fast to evaluate:

Let $\boldsymbol{W} \in \mathbb{R}^{d\times H}$, $\boldsymbol{u} \in \mathbb{R}_{>0}^H$, $\boldsymbol{b} \in \mathbb{R}_{>0}^H$ and $c \in \mathbb{R}_{>0}$. Then we define ϕ as

$$\phi(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{u}, \boldsymbol{b}, c) := \boldsymbol{u}^T \texttt{softsign}(\boldsymbol{W}\boldsymbol{x}, \boldsymbol{b}) + c \cdot \boldsymbol{x}^T\boldsymbol{x}, \tag{39}$$

where

$$\texttt{softsign}(\boldsymbol{x}, \boldsymbol{b}) := \log(\boldsymbol{b} + \cosh(\boldsymbol{x})). \tag{40}$$

Here H is a hyper-parameter that can control the complexity of the convex potential and thus its capability to model complicated multi-modal density.

Computing the volume element For $d > 3$ computing the volume element, boils down to computing

$$|\boldsymbol{J}_\Phi| = \det \boldsymbol{E}_{\boldsymbol{p}'}^T \boldsymbol{J}_\Phi(\boldsymbol{p}) \boldsymbol{E}_{\boldsymbol{p}} \tag{41}$$

by numeric means, which can become expensive and numerically unstable. For $d = 3$ we can compute the full jacobian $\boldsymbol{J}_\Phi(\boldsymbol{p})$, e.g., via `jax.jacrev` or `torch.autograd.functional.jacobian`. We can further compute the tangent bases $\boldsymbol{E}_{\boldsymbol{p}}, \boldsymbol{E}_{\boldsymbol{p}'}$ cheaply via a three-step Gram-Schmidt procedure relative to three standard basis vectors $\boldsymbol{e}_i, \boldsymbol{e}_j, \boldsymbol{e}_k$ in $\mathbb{R}^4$ which are independent of $\boldsymbol{p}$. Finally, we are only left with computing the determinant of the 3×3 matrix $\boldsymbol{A} = \boldsymbol{E}_{\boldsymbol{p}'}^T \boldsymbol{J}_\Phi(\boldsymbol{p}) \boldsymbol{E}_{\boldsymbol{p}}$ which can be done analytically and numerically stable, e.g., by computing $\det \boldsymbol{A} = (\boldsymbol{a}_0 \times \boldsymbol{a}_1)^T \boldsymbol{a}_2$.

Numerical inverse We can invert the projective convex gradient map by minimizing the residual $\ell\colon \mathbb{R}^{d+1} \to \mathbb{R}$

$$\ell(\boldsymbol{x}) = \left\| \Phi_{CG}\left(\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\right) - \boldsymbol{p}' \right\|^2 \tag{42}$$

to get $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^4} \ell(\boldsymbol{x})$ and setting $\Phi^{-1}(\boldsymbol{p}') = \frac{\boldsymbol{x}^*}{\|\boldsymbol{x}^*\|}$. For the experiments in this paper, we minimized ℓ after training using the LBFGS solver (Liu & Nocedal, 1989) as implemented in `jaxopt` (Blondel et al., 2022). For $d = 3$ and the potentials used in this work, the method converges up to an absolute error of < 0.00001 in 10-15 iterations.

Affine quaternion flows from Liu et al. (2023) are a special case Let $\boldsymbol{W} \in \mathrm{GL}(\mathbb{R}^4)$. The function $\phi(\boldsymbol{p}) = \boldsymbol{p}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{p}$ is strictly convex and satisfies all premises of the former theorem. Then

$$\Phi(\boldsymbol{p}) = \frac{\nabla_{\boldsymbol{p}}\phi(\boldsymbol{p})}{\|\nabla_{\boldsymbol{p}}\phi(\boldsymbol{p})\|} = \frac{\boldsymbol{W}\boldsymbol{p}}{\|\boldsymbol{W}\boldsymbol{p}\|} \tag{43}$$

is exactly the affine quaternion map as defined in Liu et al. (2023).

B.5. Bundle flows

Let $\pi : S^3 \to SO(3)$ denote the fiber bundle projection, with the fibers $\{-1, 1\}$. Note that this is also a covering map. Thus, for any point $p \in SO(3)$, there is an open neighbourhood $p \in U \subset SO(3)$, for which we can we can locally trivialize the fiber bundle, meaning in this case, we can pick an open set $\hat{U} \subset S^3$ and negation $-\hat{U} \subset S^3$, with two homeomorphisms $h_{\pm U} : U \xrightarrow{\sim} \pm\hat{U}$, such that $\{+\hat{U}, -\hat{U}\} = \pi^{-1}(U)$ and $\pi \circ h_{+\hat{U}} = \pi \circ h_{-\hat{U}} = \mathrm{id}_U$.

Let $\hat{f} : S^3 \to S^3$ be a sign-equivariant mapping, meaning that $-\hat{f}(x) = \hat{f}(-x)$. Then we can define a function $f : SO(3) \to SO(3)$, which restricted to any neighbourhood $U \subset SO(3)$ is defined as:

$$f\big|_U = \pi \circ \hat{f} \circ h_{+\hat{U}} = \pi \circ \hat{f} \circ h_{-\hat{U}}$$

Note that this construction does not depend on the choice of trivialization, and is therefore well-defined. Then following diagram commutes, making the pair $(\hat{f}, f)$ a fiber bundle morphism.

$$\begin{tikzcd} S^3 \arrow[r, "\hat{f}"] \arrow[d, "\pi"'] & S^3 \arrow[d, "\pi"] \\ SO(3) \arrow[r, "f"] & SO(3) \end{tikzcd} \tag{44}$$

For a measurable space X, let PX denote the space of measures on that space. For a measurable map $a : X \to Y$, let $a^* : PX \to PY$ denote the pushforward. For a stochastic map $b : X \to PY$, we also denote by $b^* : PX \to PY$ the induced map between measures.

Now, let $s : SO(3) \to PS^3$ be a stochastic map that is a stochastic section to the projection: $\pi^* \circ s^* = \mathrm{id}_{PSO(3)}$. Then consider the following diagram:

$$\begin{tikzcd} P\,SO(3) \arrow[r, "s^*"] \arrow[dr, "\mathrm{id}"'] & PS^3 \arrow[r, "\hat{f}^*"] \arrow[d, "\pi^*"'] & PS^3 \arrow[d, "\pi^*"] \\ & P\,SO(3) \arrow[r, "f^*"'] & P\,SO(3) \end{tikzcd}$$

By the assumption that s is a section to π^*, the left triangle commutes. The right square commutes because diagram (44) commutes, which induces a commuting diagram of push-forwards. As both the left triangle and the right square commute, the outer two paths commute. This implies that the push-forward given by $f : SO(3) \to SO(3)$ equals first stochastically lifting to S^3, then applying the sign-equivariant $\hat{f} : S^3 \to S^3$, and projecting back to $SO(3)$. That computation does not depend on the choice of the particular stochastic section $s : SO(3) \to PS^3$. One choice could be to deterministically choose either point in each fiber.

More generally, this construction works for any bundle, if the map on the total space is a bundle morphism (satisfies diagram (44)) and we can construct a stochastic section to the projection map.

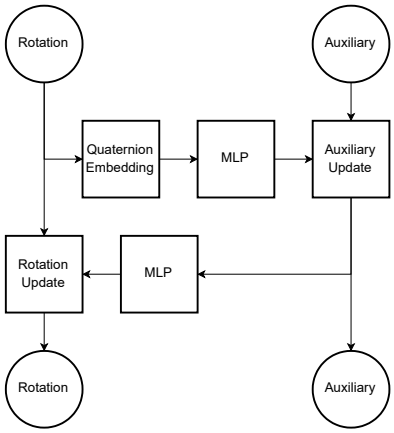

Figure 4. The coupling used for the tetrahedron experiment.

C. Details on tetrahedron experiments

Control parameters of the chosen force-field The force field is given by the potential

$$u(\boldsymbol{x}) = C \cdot \sum_{k=1}^{5} \sum_{d=1}^{3} (x_{kd} - c_d)^4, \tag{45}$$

with control parameters $\boldsymbol{c} = (0.09, -0.073, 0.), C = 136.98630$.

Setup of the simulation and data generation We use a methane molecule with reference coordinates given as

	x	y	z
C	-0.037	0.090	0.000
H1	0.070	0.090	0.000
H2	-0.073	0.012	0.064
H3	-0.073	0.073	-0.100
H4	-0.073	0.184	0.035

We then enforce the position of the carbon to be fixed by putting a position restraint to it. We fix the inner degrees of freedom by adding a bond constraint to the `CH` bonds and angle restraints to all possible `HCH` angles, fixing them to `109.47122°`.

We then run an OpenMM simulation using a Langevin integrator at 100K. We chose a time step of 1ps and only keep each 500th frame as a sample to ensure proper mixing. This results in a dataset of 50,000 samples.

Flow model We use augmented normalizing flows (Huang et al., 2020; Chen et al., 2020) and add auxiliary noise dimensions to our data as otherwise our dataset would only consist of one quaternion and as such could not be modeled with coupling layers. We use a two-dimensional unit normal distribution to model the auxiliary noise in data and latent space. We furthermore use an uninformed Von-Mises-Fisher (VMF) density with concentration parameter 2.5 as base density for the rotations. To satisfy flip-invariance, we model this base density as a mixture of the location and its antipode. We then set up a two-layer coupling flow, coupling the rotation degree of freedom and the noise as depicted in Fig. 4.

The conditioning functions producing the parameters of the flows are simple two-layer dense nets with 128 hidden units and GELU activation (Hendrycks & Gimpel, 2016). To ensure flip-invariance for the parameters of the auxiliary transformation we embed the conditioning quaternions using the following self-attention mechanism before feeding them into the dense nets (see Fig. 5):

- Let $S\colon \mathbb{R}^4 \to \mathbb{R}$ and $F\colon \mathbb{R}^4 \to \mathbb{R}^H$ be linear layers where H is some embedding dimension.

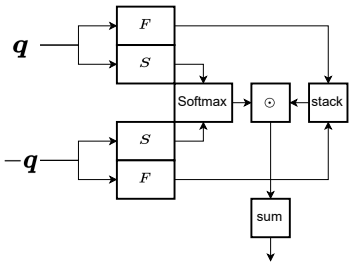

Figure 5. The flip-invariant embedding is used for the quaternions before feeding them into conditioning NN. The layers F and S are shared over inputs.

- Then we compute the quaternion embedding as

$$G(\boldsymbol{q}) = \sum_{s\in\{-1,1\}} \frac{\exp(S(s\cdot\boldsymbol{q}))}{\exp(S(s\cdot\boldsymbol{q})) + \exp(S(-s\cdot\boldsymbol{q}))} \cdot F(s\cdot\boldsymbol{q}) \tag{46}$$

For the auxiliary transformations we rely on simple real-NVP blocks (Dinh et al., 2017). For the rotation transformation, we tried the following setups:

- the affine flows of Liu et al. (2023).
- the symmetrized Moebius layers as presented in this work.
- three variants of the projective convex gradient maps as presented in this work using the potential defined in Eq. 39 setting $H = 8, 32,$ and 128 respectively.

Fig. 2 in the main text shows the result for the convex potential with $H = 128$. We show a comprehensive ablation of how the quality degrades when varying H in figure 6.

Training By augmenting the data $\mu(\boldsymbol{q})$ distribution with auxiliary noise $\boldsymbol{z}$, we obtain a new *augmented data* density

$$\mu(\boldsymbol{q},\boldsymbol{z}) = \mu(\boldsymbol{q}) \cdot \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I}) \,. \tag{47}$$

To match dimensionality, we furthmore have to augment the base density as well, giving us

$$\mu_0(\boldsymbol{q},\boldsymbol{z}) = \mathrm{VMF}(\boldsymbol{e}_0,\kappa) \cdot \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I}) \,. \tag{48}$$

Here $\boldsymbol{e}_0 = (1,0,0,0)$ and $\kappa = 2.5$.

Then the likelihood objective in eq. 5 transforms into

$$\boldsymbol{\theta}_{ML} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{q},\boldsymbol{z}\sim\mu(\boldsymbol{q},\boldsymbol{z})} \left[-\log \Phi(\mu_0)(\boldsymbol{q},\boldsymbol{z};\boldsymbol{\theta})\right] . \tag{49}$$

We optimize this objective for $50,000$ steps for each candidate flow. We used the ADAM optimizer (Kingma & Ba, 2015) with a batch size of 32 and a learning rate of 0.0005.

D. Details on the Ice XI experiments

We considered 3 different setups (see also Fig. 3), in each case the temperature of the base distribution was T_0 250 K. We varied the number of water molecules N and the temperature of the target distribution T:

a) N = 16, T = 100 K

b) N = 16, T = 50 K

c) N = 128, T = 100 K

Setup (a) and (b) use the same base distribution.

Details of the simulations and data generation process The initial configuration of ice XI was generated with the GenIce2 software (Matsumoto et al., 2021), using a single cell for the $N = 16$ system and two cells per dimension for the $N = 128$ one. We did not change the default size of the generated box. We run molecular dynamics simulations with the OpenMM library (Eastman et al., 2017) using the TIP4P-Ew rigid water force-field (Horn et al., 2004) (cutoff length half the smaller box edge with a switching function for smooth interactions), and Langevin middle integrator (Zhang et al., 2019) with integration step of 1 fs and friction coefficient of 1 ps^{-1}. We chose the TIP4P-Ew water model because it is readily available in OpenMM, but similar results can be obtained with other rigid water models, such as TIP4P-ice (Abascal et al., 2005). We run iterations of 500 MD steps and store a molecular configuration at each iteration. All our MD simulations start with an equilibration run of 10,000 iterations, which is then discarded.

To generate the data used for training we run a MD simulation of 10,000 iterations for each of the two base distributions, and another 10,000 iterations MD for each of the two evaluation sets.

We note that ice XI is likely only metastable at the thermodynamic conditions that we consider, however, it is stable enough that we can run long MD simulations without observing any phase transitions, and thus it is perfectly suitable for our purposes.

Flow model For all three experiments, we used the same coupling architecture, where we couple positions and rotations in a round-robin way over four iterations (see Fig. 7).

The system is invariant with respect to global translations, thus the number of degrees of freedom associated to the positions of the molecules is not $3N$ but $3(N-1)$. To account for this, we fix the position of one of the molecules (but not the rotation) and apply the flow only to the remaining $N-1$ (Wirnsberger et al., 2022).

To share parameters and reuse local substructure, we model the conditioning networks with transformers (Vaswani et al., 2017) using multi-headed self-attention (see Figs. 8, 9). Since atoms in the crystal phase have well-specified positions we do not need to enforce strict permutation symmetry, e.g., as you would need to do when studying liquids. Similar to positional encoding in language models, we encode the molecule index as a one-hot vector before feeding it into the first transformer block. As we furthermore have to guarantee flip-invariance for the position conditioner, we use a modified attention mechanism where we stack rotations and the features of the last layer into different heads and then run the rotation logits through a square function to cancel its sign (see Fig. 10 a)). In all experiments, we use two transformer blocks per flow layer each using 8 heads and 32 channels.

For updating the positions conditioned on the rotations we were using Moebius layers (Rezende et al., 2020). For updating the rotations conditioned on the positions we used the symmetrized Moebius transforms described in the main text.

The flow is initialized to be the approximately the identity, so that at the beginning of training the mapped configurations have reasonable energies. This is obtained by multiplying all the parameters of the coupling layers with a sigmoid function whose arguments are also learnable parameters.

The total number of trainable parameters is $290,896$ for system (a) and (b), and $7,458,896$ for system (c).

Training We train each model using ADAM (Kingma & Ba, 2015) for 1000 iterations per epoch over 10 epochs. We use a batch size of 32 and a cosine scheduler for the learning rate, which annealed the learning rate from 1e-3 in the first epoch to 1e-5 in the final epoch. We used the same training setup for all the models.

Details on the evaluation Once the flow is trained, we apply it to the evaluation dataset to obtain the potential energy profiles and the oxygen-oxygen radial distribution functions shown in Fig. 3. To estimate the uncertainty for the LFEP estimator we use the bootstrapping method, and compute it 10 times by subsampling the evaluation data from the base distribution.

Similarly, we estimate the Kish effective sampling size (Kish, 1965) of the generated samples from 10,000 random points

of the evaluation dataset. The obtained averages and standard deviations over 10 estimates are: (a) $22.74 \pm 1.58\%$, (b) $6.90 \pm 1.36\%$, (c) $0.39 \pm 0.06\%$.

The data for the reference MBAR ΔF calculations are obtained by running 10,000 MD iterations at various intermediate temperatures between the base and the target. In total, we performed 5 MD runs for setup (a), 10 for (b), and 10 for (c). The temperature ladder follows a geometric distribution. The uncertainty over the MBAR calculations is estimated with bootstrapping, using the `pymbar` package (`https://github.com/choderalab/pymbar`).

Examples of the sampled atomistic configurations are shown in Fig. 11, as 2D projections.

Computational cost We did not optimise the computational efficiency of either our method or the MBAR reference, as this was not the aim of this work. We expect the use of normalizing flows to bring a clear speed-up over more traditional MD methods, in cases where energy evaluation is more expensive, such as with interatomic potentials based on neural networks or on higher levels of theory. In our experiment, the number of energy evaluations used for the MBAR estimate of (b) and (c) is $10 \times 10,000 \times 500 = 50,000,000$, while for the LFEP estimate is $2 \times 10,000 \times 500 = 10,000,000$ for the MD sampling of the base distribution, plus $10 \times 1000 \times 32 = 320,000$ for the training. Training on a GeForce GTX 1080 Ti took about 10 minutes for each of the small systems and 30 minutes for the larger one.

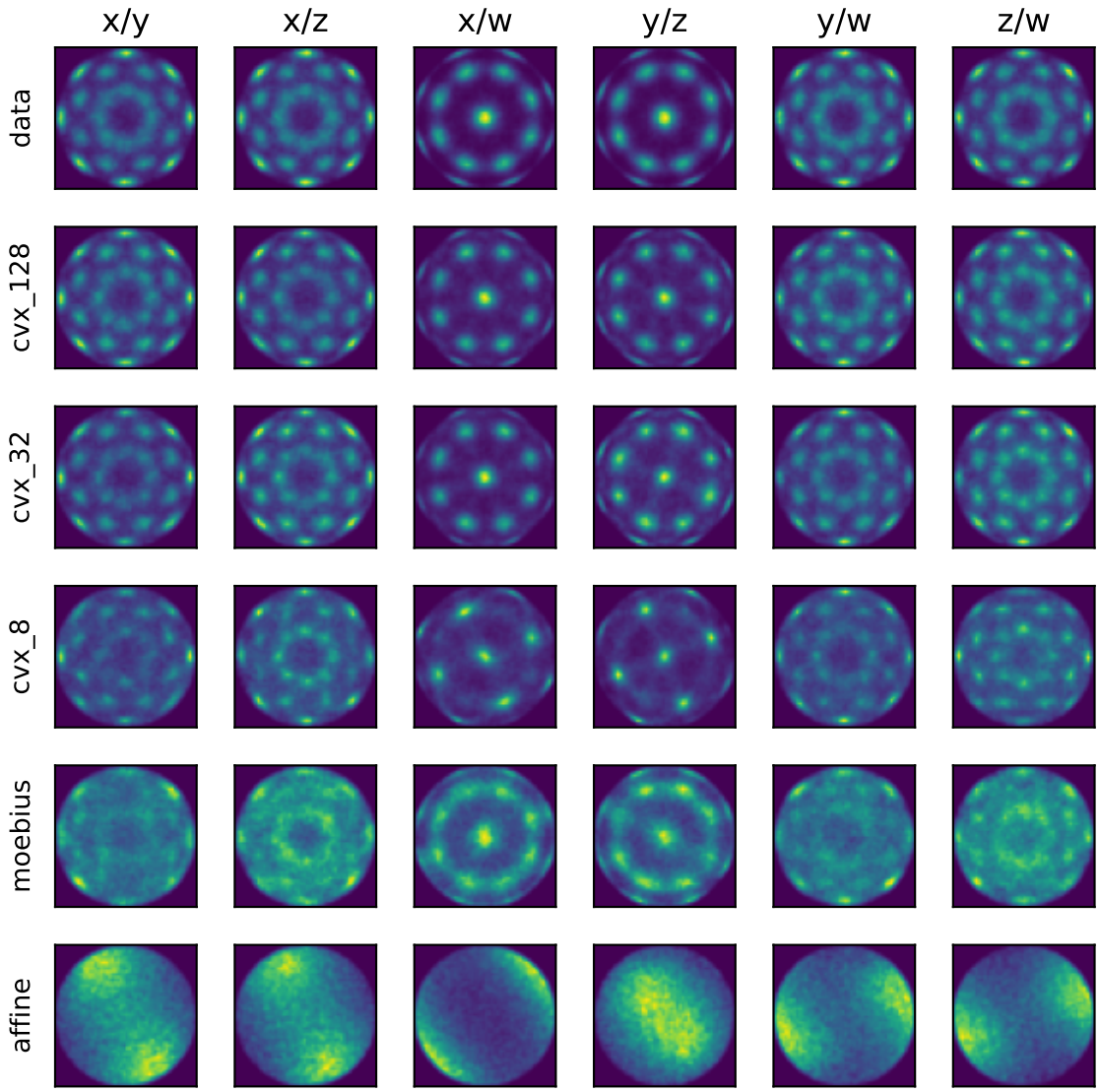

Figure 6. Full ablation for the methane experiment: in addition to the results shown in Fig. 2 we show how the performance of the projective convex gradient map degrades by varying H from $H = 8$ to $H = 128$.

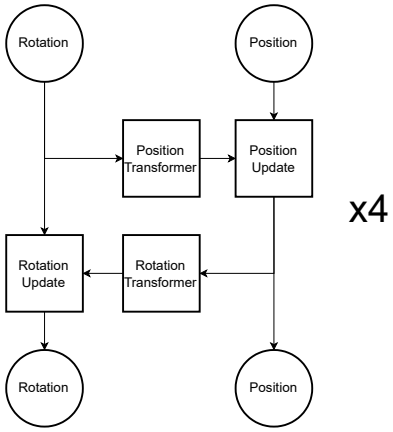

Figure 7. The flow architecture used for the ice XI experiments. The coupling layers are repeated four times.

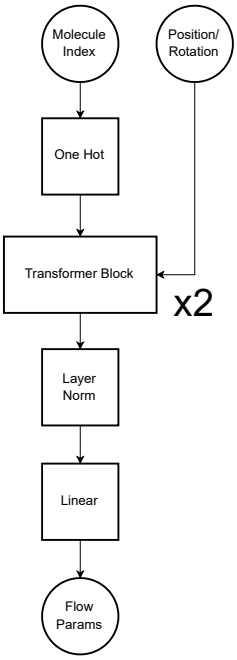

Figure 8. The conditioning layers for both the position and rotation conditioner. We use positional encoding and first embed the molecule index as a one-hot vector. Then we proceed with a stack of two transformer blocks. Finally, we decode the output into the flow parameters.

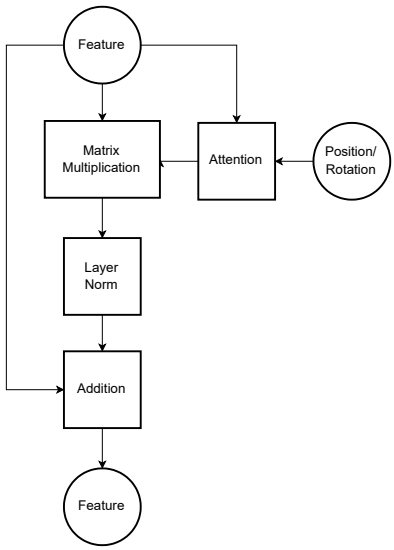

Figure 9. The transformer blocks follow the same structure for both positions and rotations and only differ in the attention mechanism used. As in usual transformer models, the features of the last layer together with current positions/rotations determine the self-attention matrix. We then multiply the last features with the computed attention matrix and add the result to the features of the last layer.

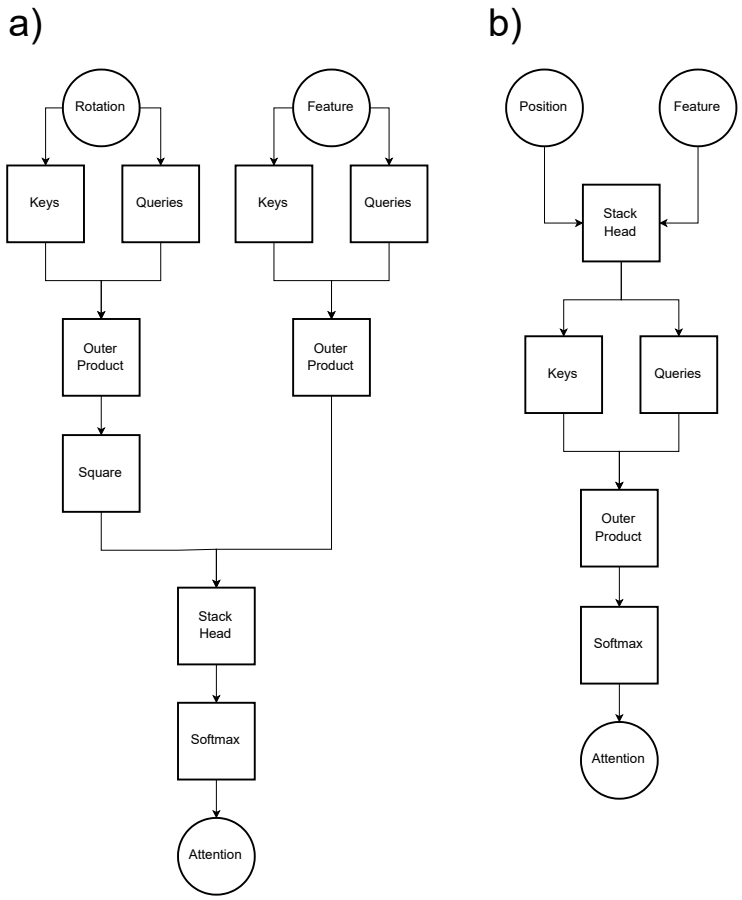

Figure 10. The attention mechanisms used within the transformer blocks. a) The attention mechanism for the position transformer: to preserve flip-invariance, we separate the information flow from rotations and features into separate heads. Then we apply a square function on the rotation logits to cancel the sign. Finally, we can stack along the head dimension and proceed with the softmax as usual. b) The attention mechanism for the rotation transform boils down to a usual self-attention block, where we first concatenate features and positions before sending them through key and query blocks.

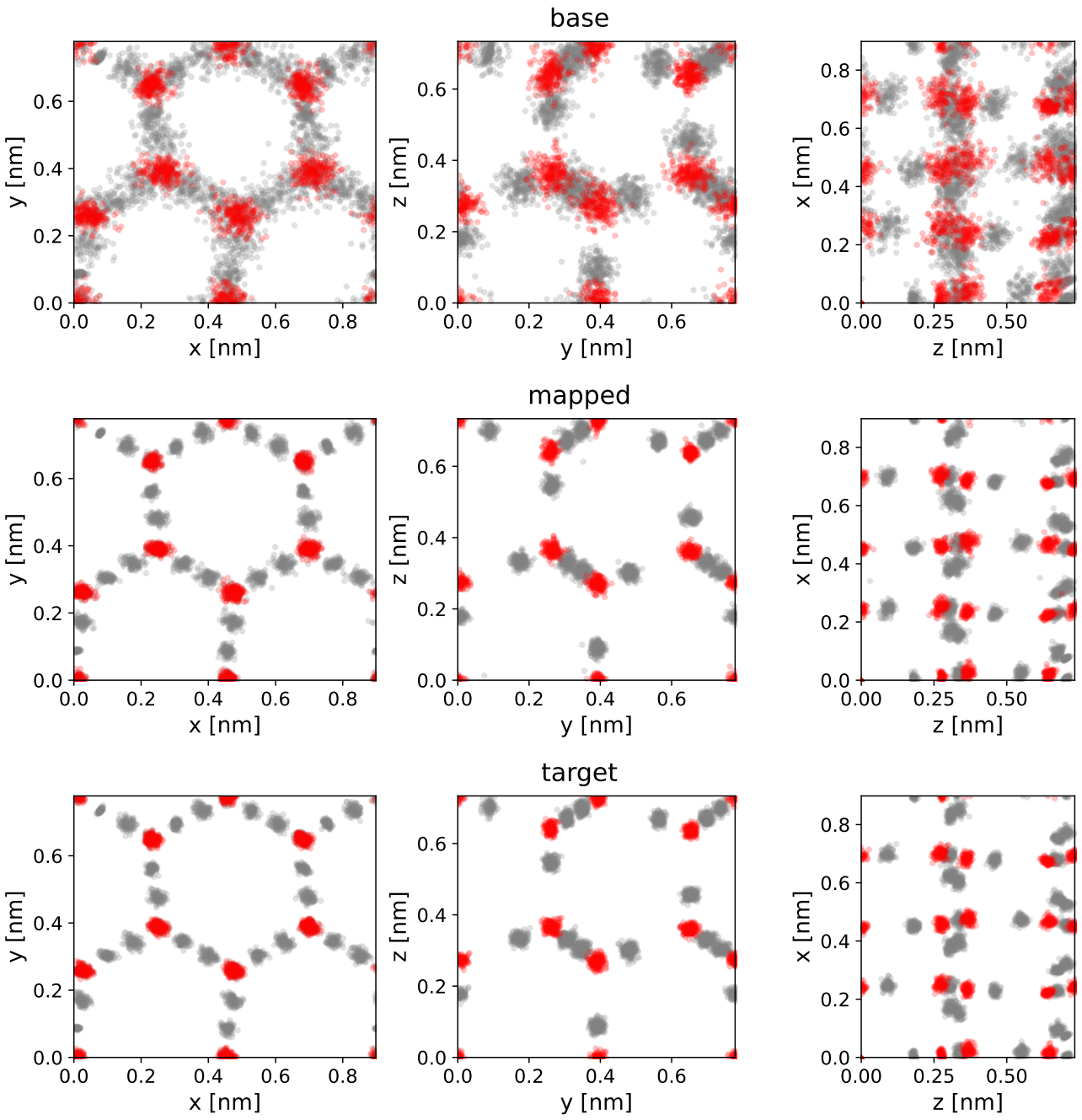

Figure 11. Atomistic configurations for the case of N = 16. Each row shows the 2D projection of 100 different samples, with the oxygens colored in red and the hydrogens in gray. The first row contains samples from the base distribution at T_0 = 250 K, the second row shows how those configurations are mapped to the target by the NF, and the third row shows MD samples from the target distribution at T = 50 K. As the flows were purely trained on energies, these target samples were not seen during the training.