Equivariant Flows: exact likelihood generative learning for symmetric densities.

Jonas Köhler* 1 Leon Klein* 1 Frank Noé 1 2 3

1 Freie Universität Berlin, Department of Mathematics and Computer Science. 2 Freie Universität Berlin, Department of Physics. 3 Rice University, Department of Chemistry. Correspondence to: Jonas Köhler <jonas.koehler@fu-berlin.de>, Frank Noé <frank noe@fu-berlin.de>.

Abstract

Normalizing flows are exact-likelihood generative neural networks which approximately transform samples from a simple prior distribution to samples of the probability distribution of interest. Recent work showed that such generative models can be utilized in statistical mechanics to sample equilibrium states of many-body systems in physics and chemistry. To scale and generalize these results, it is essential that the natural symmetries in the probability density – in physics defined by the invariances of the target potential – are built into the flow. We provide a theoretical sufficient criterion showing that the distribution generated by equivariant normalizing flows is invariant with respect to these symmetries by design. Furthermore, we propose building blocks for flows which preserve symmetries which are usually found in physical/chemical many-body particle systems. Using benchmark systems motivated from molecular physics, we demonstrate that those symmetry preserving flows can provide better generalization capabilities and sampling efficiency.

1. Introduction

Generative learning using exact-likelihood methods based on invertible transformations has had remarkable success in accurately representing distributions of images (Kingma & Dhariwal, 2018), audio (Oord et al., 2017) and 3D point cloud data (Liu et al., 2019b; Noé et al., 2019).

Recently, Boltzmann Generators (BG) (Noé et al., 2019) have been introduced for sampling Boltzmann type distributions \( \rho'(x) \propto \exp(-u(x)) \) of high-dimensional many-body problems, such as valid conformations of proteins. This approach is widely applicable in the physical sciences, and has also been employed in the sampling of spin lattice states (Nicoli et al., 2019; Li & Wang, 2018) and nuclear physics models (Albergo et al., 2019). In contrast to typical generative learning problems, the target density \( \rho'(x) \) is specified by definition of the many-body energy function \( u(x) \) and the difficulty lies in learning to sample it efficiently. BGs do that by combining an exact-likelihood method that is trained to approximate the Boltzmann density \( \rho'(x) \), and a statistical mechanics algorithm to reweigh the generated density to the target density \( \rho'(x) \).

In order to make further progress, it is essential to develop exact-likelihood generative models that respect the symmetries of \( u(x) \), for example invariance of the energy with respect to global rotation or permutations of identical particles. In this work, we pursue this goal by making the following contributions:

- A sufficient criterion and a construction principle to construct exact-likelihood generative models for symmetric densities over \( \mathbb{R}^n \) based on equivariant flows.
- A numerically efficient implementation of the framework for symmetric many-body particle systems.
- An efficient dynamics function based on Gaussian kernels, which can be use for continuous normalizing flows, that is easy to regularize, parameter-efficient and allows a fast and analytic divergence computation.
- Empirical evidence, that such symmetric flows provide better generalization in the case of stiff many-body systems with symmetric energies, compared to non-symmetric approaches or classic sampling.

2. Related Work

Statistical mechanics The workhorse for sampling Boltzmann-type distributions \( p(x) \propto \exp(-u(x)) \) with known energy function \( u(x) \) are Molecular dynamics (MD) and Markov-Chain Monte-Carlo (MCMC) simulations. MD and MCMC take local steps in configurations \( x \), are guaranteed to sample from the correct distribution for infinitely long trajectories, but are subject to the rare event sampling problem, i.e. the get stuck in local energy minima of \( u(x) \) for long time. Statistical mechanics has developed many
tools to speed up rare events by adding a suitable bias energy to \( u(x) \) and subsequently correcting the generated distribution by reweighing or Monte-Carlo estimators using the ratio of true over generated density, e.g. (Torrie & Valalleau, 1977; Bennett, 1976; Laio & Parrinello, 2002; Wu et al., 2016). These methods can all speed up MD or MCMC sampling significantly, but here we pursue sampling of the equilibrium density with flows.

### Normalizing flows

Normalizing flows (NFs) are diffeomorphisms \( f_\theta : \mathbb{R}^n \to \mathbb{R}^n \) which transform samples \( z \sim \rho \) from a simple prior density \( \rho \) into samples \( x = f_\theta(z) \) (Tabak et al., 2010; Tabak & Turner, 2013; Rezende & Mohamed, 2015; Papamakarios et al., 2019). Denoting the density of the transformed samples \( \rho_\theta \), we obtain the probability density of any generated point via the change of variables equation

\[
\rho_\theta(x) = \rho \left( f_\theta^{-1}(x) \right) \det \frac{\partial f_\theta^{-1}(x)}{\partial x}.
\]

\( \rho_\theta \) is also called the push-forward of \( \rho \) along \( f_\theta \).

While flows can be used to build generative models by maximizing the likelihood on a data sample, having access to tractable density is especially useful in variational inference (Rezende & Mohamed, 2015; Tomczak & Welling, 2016; Louizos & Welling, 2017; Berg et al., 2018) or approximate sampling from distributions given by an energy function (Oord et al., 2017), which can be made exact using importance sampling (Müller et al., 2018; Noé et al., 2019).

The majority of NFs can be categorized into two families: 1) Coupling layers (Dinh et al., 2014; 2016; Kingma & Dhariwal, 2018; Müller et al., 2018), which are a subclass of autoregressive flows (Germain et al., 2015; Papamakarios et al., 2017; Huang et al., 2018; De Cao et al., 2019; Durkan et al., 2019), and 2) residual flows (Chen et al., 2018; Zhang et al., 2018; Grathwohl et al., 2018; Behrmann et al., 2018; Chen et al., 2019).

Symmetries in flow models have been discussed in the context of permutations in graphs (Liu et al., 2019a). A preliminary account of equivariant normalizing flows has been given in two recent workshop submissions (Rezende et al., 2019; Köhler et al., 2019).

### Boltzmann-generating flows

While flows and other generative models are typically used for estimating the an unknown density \( \rho' \) from samples and then generating new samples from it, BGs know the desired target density \( \rho'(x) \propto \exp(-u(x)) \) up to a prefactor and aim at learning to efficiently sample it (Noé et al., 2019).

A BG combines two elements to achieve this goal:

1. An exact-likelihood generative model that generates samples \( x_k \) from a density \( \rho_{f_\theta} \) that approximates the given Boltzmann-type target density \( \rho' \).
2. An algorithm to reweigh the generated density to the target density \( \rho' \). For example, using importance sampling the asymptotically unbiased estimator of the expectation value of observable \( O(x) \) is:

\[
\mathbb{E}_{x \sim \rho'}[O] \approx \frac{\sum_k w(x_k)O(x_k)}{\sum_k w(x_k)}, \quad x_k \sim \rho_{f_\theta},
\]

where the importance weights

\[
w(x_k) = \exp(-u(x_k))/\rho_{f_\theta}(x_k)
\]

can be computed from the trained flow.

The exact likelihood model is needed in order to be able to conduct the reweighing step. When a flow is used in order to generate asymptotically unbiased samples of the target density, we speak of a Boltzmann-generating flow.

Boltzmann-generating flows are trained to match \( \rho_{f_\theta} \approx \rho' \) using loss functions that also appear in standard generative learning problems, but due to the explicit availability of \( \exp(-u(x)) \) their functional form and interpretation changes:

1. **KL-training** We minimize the reverse Kullback-Leibler divergence \( K L(\rho_{f_\theta} \parallel \rho') \):

\[
\mathcal{L}_{KL} = \mathbb{E}_{x \sim \rho}[u(f_\theta(z)) - \log \det \frac{\partial f_\theta(z)}{\partial z}]
\]

This approach is also known as energy-based training where the energy corresponding to the generated density is matched with \( u(z) \).

2. **ML-training**: If data \( \{x_n\}_{n=1}^N \) from a data distribution \( \rho'_{\text{data}} \) is given that at least represents one or a few high-probability modes of \( \rho' \), we can maximize the likelihood under the model, as is typically done when performing density estimation.

\[
\mathcal{L}_{ML} = \mathbb{E}_{x \sim \rho'_{\text{data}}}[\log \rho(f_\theta^{-1}(x)) - \log \det \frac{\partial f_\theta^{-1}(x)}{\partial z}]
\]

The final training loss is then obtained using a convex sum over both losses, where the mixing parameter \( \lambda \) may be changed from 0 to 1 during the course of training

\[
\mathcal{L} = (1 - \lambda)\mathcal{L}_{ML} + \lambda\mathcal{L}_{KL}
\]

### 3. Invariant densities via equivariant flows

In this work we consider densities \( \rho, \rho' \) over euclidean vector spaces \( \mathbb{R}^n \) which are invariant w.r.t. to symmetry transformations e.g. given by rotations and permutations of the
Equivariant Flows: exact likelihood generative learning for symmetric densities.

space. In other words, we want to construct flows such that both, the prior and the target density share the same symmetries.

More precisely, let \( G \) be a group which acts on \( \mathbb{R}^n \) via a representation \( R: G \to GL(n) \), \( g \to R_g \) and assume that \( \rho \) is \( G \)-invariant w.r.t. \( G \), i.e. \( \forall g \in G, x \in \mathbb{R}^n : \rho(R_g x) = \rho(x) \).

Then we first remark that for any \( g \in G \) the matrix \( R_g \) satisfies \( \det(R_g) \in \{-1, 1\} \). This allows us to formulate our first theoretic result:

**Theorem 1.** Let \( \rho \) be a density on \( \mathbb{R}^n \) which is \( G \)-invariant and \( G \supset H \). If \( f \) is a \( H \)-equivariant diffeomorphism, i.e. \( \forall h \in H, x \in \mathbb{R}^n : f(R_h x) = R_h f(x) \), then \( \rho \) is \( H \)-invariant.

As a direct consequence, any push-forward of an isotropic normal distribution along a \( H \)-equivariant diffeomorphism will result in a \( H \)-invariant proposal density

4. Constructing equivariant flows

In general it is not clear how to define equivariant diffeomorphisms which provide tractable inverses and Jacobians. We will provide a possible implementations based on the recently introduced framework of continuous normalizing flows (CNFs) (Chen et al., 2017).

**Equivariant dynamical systems** CNFs define a dynamical system via a time-dependent vector field \( v: \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^n \). If \( v \) is globally Lipschitz, we can map each \( z \in \mathbb{R}^n \) onto the unique characteristic function \( x_{v,z}: [0, \infty) \to \mathbb{R}^n \), which solves the Cauchy-problem

\[
\frac{d}{dt} x(t) = v(x_{v,z}(t), t), \quad x_{v,z}(0) = z.
\]

This allows us to define a bijection \( F_{v,T}: \mathbb{R}^n \to \mathbb{R}^n \) for each \( T \in [0, \infty) \) by setting

\[
F_{v,T}(z) = x_{v,z}(0) + \int_0^T dt \, v(x_{v,z}(t), t).
\]

Given a density \( \rho \) on \( \mathbb{R}^n \), each \( T \) defines a push-forward \( \rho_{F_{v,T}} \) along \( F_{v,T} \), which satisfies the continuous change of variable

\[
\frac{d}{dt} \log \rho_{F_{v,T}}(x_{v,z}(t)) = -\text{div} (v(x_{v,z}(t), t)).
\]

By following the characteristic this allows to compute the total density change as

\[
\log \frac{\rho_{F_{v,T}}(x_{v,z}(T))}{\rho(x_{v,z}(0))} = -\int_0^T dt \, \text{div} (v(x_{v,z}(t), t)).
\]

Using CNFs, equivariant flows can be constructed very naturally:

\footnote{All proofs and derivations can be found in the Suppl. Material.}

**Theorem 2.** Let \( v \) be a \( H \)-equivariant vectorfield on \( \mathbb{R}^n \) (not necessarily bijective). Then for each \( T \in [0, \infty) \) the bijection \( F_{v,T} \) is \( H \)-equivariant.

Consequently, if \( \rho \) is a \( G \)-invariant density on \( \mathbb{R}^n \) and \( G \supset H \), then each push-forward \( \rho_{F_{v,T}} \) is \( H \)-invariant.

**Equivariant gradient fields** There has been a significant amount of work in recent years proposing \( G \)-equivariant flows for different groups acting on \( \mathbb{R}^n \). A generic implementation however is given by a gradient flow: if \( \Phi: \mathbb{R}^n \to \mathbb{R} \) is a \( G \)-invariant function, the vector \( \nabla_x \Phi \) will transform \( G \)-equivariantly.

Gradient flows (not necessarily \( G \)-equivariant) can map any \( \rho \) onto any \( \rho' \) over \( \mathbb{R}^n \) as long as both densities do not vanish (Benamou & Brenier, 2000; McCann, 2001) and have been discussed in the context of density estimation (Zhang et al., 2018; Papamakarios et al., 2019).

**Numerical implementations** While providing an elegant solution, implementing equivariant flows using continuous gradient flows is numerically challenging due to three aspects.

First, even if \( F_{v,T} \) is invertible assuming exact integration, there are no such guarantees for any discrete-time approximation of the integral, e.g. using Euler or Runge-Kutta integration. Thus, Chen et al. propose adaptive-step solvers, such as Dopr5 (Dormand & Prince, 1980), which can require hundreds of vector field evaluations to reach a satisfying numerical accuracy.

Second, in order to train \( v \) via the adjoint method as suggested by Chen et al., gradients of the loss w.r.t. parameters are obtained via backward integration. However, in general, there are no guarantees that this procedure is stable, which therefore can result in very noisy gradients, leading to long training times and inferior final results (Gholami et al., 2019). In contrast to this optimize-then-discretize (OTD) approach, Gholami et al. suggest to unroll the ODE into a fixed-grid sequence and backpropagate the error using classic automatic differentiation (AD). Such a discretize-then-optimize (DTO) approach will guarantee that gradients are computed correctly, but might suffer from inaccuracy due to the discretization errors as mentioned before. Throughout our experiments, we rely on the latter approach during training and show that for our presented architecture OTD and DTO will yield similar results, while the latter offers a significant speedup per iteration, more robust training and faster convergence.

Finally, computing the divergence of \( v \) using off-the-shelf AD frameworks requires \( O(n) \) backpropagation passes, which would result in an infeasible overhead for high-dimensional systems (Grathwohl et al., 2018). Thus, Grath-
wohl et al. suggest an approximation via the Hutchinson-estimator (Hutchinson, 1989). This is an unbiased rank-1 estimator of the divergence where variance scales with \(O(n)\). As we show in our experiments, even for small particle systems, relying on such an estimator will render importance weighing and thus the benefits of Boltzmann generating flows useless. Another approach relies on designing special dynamics functions, in which input dimensions are decoupled and then combine the \(\text{detach}\)-operator with one back-propagation pass to compute the divergence exactly (Chen & Duvenaud, 2019). For general symmetries as studied in weighing and thus the benefits of Boltzmann generating flows, however, in order to compute unbiased estimates of target densities which is essential for physics applications, a variational approximation of \(\rho'(p|q)\) cannot be applied.

5. Sampling of coupled particle systems

We apply our framework on the problem of sampling systems \(x \in \mathbb{R}^n, \ n = N \cdot D\) consisting of \(N\) particles \(x_i\) with \(D \in [2,3]\) degrees of freedom, which are coupled via a potential energy \(u(x)\). In thermodynamic equilibrium such a system follows a Boltzmann-type distribution \(\rho'(x) \propto \exp(-u(x))\). Assuming interchangeable particles in vacuum without external field, we obtain three symmetries (S1-3): \(u\) (and thus \(\rho'\)) does not change if we permute particles (S1), rotate the system around the center of mass (CoM) (S2), or translate the CoM by an arbitrary vector (S3).

Due to the simultaneous occurrence of (S1) and (S2) no autoregressive decomposition / coupling layer can be designed to be equivariant. Either a variable split has to be performed among particles or among spatial coordinates, which will break permutation and rotation symmetry respectively. Thus, residual flows are the only class of flows which can be applied here. In this work we will rely on CNFs, design an equivariant vector field, and then combine theorems 1 and 2 to conclude the symmetry of the proposal density.

Invariant prior density We first start by designing an invariant prior. By only considering systems with zero CoM symmetry (S3) is easily satisfied. The set of CoM-free systems forms a \((N-1) \cdot D\)-dimensional linear subspace \(U < \mathbb{R}^n\). Equipping \(\mathbb{R}^n\) with an isotropic normal density \(\rho\), implicitly equips \(U\) with a normal distribution \(\tilde{\rho}\). We can sample it, by sampling \(z \sim \rho\) and projecting on \(U\), and evaluate its likelihood for \(z \in U\), by computing \(\rho(z)\).

Equivariant vector field We design our vector field as a kernel dynamics such that each particle \(x_i\) is updated in the context of another particle \(x_j\) according to a time-dependent vector which acts in distance direction

\[
v_{ij}(x(t), t) = \phi(d_{ij}(t), t) \cdot r_{ij}(t) = R(t)^T W K(d_{ij}(t)) r_{ij}(t),
\]

with \(r_{ij}(t) = x_i(t) - x_j(t), \ d_{ij}(t) = \|r_{ij}(t)\|\). Here \(K : \mathbb{R} \rightarrow \mathbb{R}^K\) and \(R : \mathbb{R} \rightarrow \mathbb{R}^K\) are vector-valued functions, where each component is given by a Gaussian RBF and \(W \in \mathbb{R}^{T \times K}\) is a trainable weight matrix.
Using these forces, a particle update in time is computed as
\[
\frac{\partial x_i(t)}{\partial t} = v_i(x(t), t) = \sum_j v_{ij}(x(t), t).
\]
This update is equivariant with respect to global rotations and permutations of particles. In fact it can be interpreted as an equivariant gradient field. Furthermore, it maps mean-free systems on mean-free updates and thus satisfies the requirement of the former paragraph (see Figure 1 for architecture).

Using this architecture, the divergence becomes:
\[
\text{div} \frac{\partial x(t)}{\partial t} = \sum_{ij} \frac{\partial \phi(d_{ij}(t), t)}{\partial d_{ij}(t)} d_{ij}(t) + n \cdot \phi(d_{ij}(t)),
\]
which for the given choice of \( \phi \) can be computed exactly and as one vectorized operation reusing most of the results, which were required to compute the updates.

During training we optimize \( W \) and RBF means and bandwidths simultaneously. By keeping weights small and bandwidths large we can control the complexity of the dynamics. As we show in our experiments even a small amount of weight-decay is sufficient to properly optimize the flow with a fixed-grid solver introducing a negligible amount of error during the integration.

**6. Benchmark systems**

We study two systems where all symmetries (S1), (S2), (S3) are present

**7. Experiments**

We show the numerical accuracy and efficiency of the proposed framework in four experiments.

**7.1. Computation of divergence**

Here we show that fast and exact divergence computations are critical especially when the number of particles grows. We compare different ways to estimate the change of log-density: (1) using brute-force computation relying on AD (2) using the Hutchinson estimator described by Grathwohl (LJ) potential with parameters

\[
u^{*}(x) = \frac{1}{\tau} \sum_{i,j} a (d_{ij} - d_0) + b (d_{ij} - d_0)^2 + c (d_{ij} - d_0)^4
\]

for \( D = 2 \), which produces two distinct low energy modes separated by an energy barrier. By coupling multiple particles with such double-well interactions we can create a frustrated system with multiple metastable states. Here \( a, b, c \) and \( d_0 \) are chosen design parameters of the system and \( \tau \) the dimensionless temperature.

**LJ-13** The second system is given by the Lennard-Jones (LJ) potential with \( N = 13 \), \( D = 3 \). LJ is a model for solid-state models and rare gas clusters. LJ clusters have complex energy landscapes whose energy minima are difficult to find and sample between. These systems have been extensively studied (Wales & Doye, 1997) and are good candidates for benchmarking structure generation methods. In order to prevent particles to dissociate from the cluster at the finite sampling temperature, we add a small harmonic potential to the CoM. The LJ potential with parameters \( \epsilon \) and \( r_m \) at dimensionless temperature \( \tau \) is defined by

\[
\frac{u^{*}(x)}{\tau} = \frac{\epsilon}{\tau} \left[ \sum_{i,j} \left( \frac{r_m}{d_{ij}} \right)^{12} - 2 \left( \frac{r_m}{d_{ij}} \right)^{6} \right]
\]

**Figure 2.** The two model systems used in these experiment. Shown are the energy contributions per distance a) for the double-well and b) the Lennard-Jones potential.
Equivariant Flows: exact likelihood generative learning for symmetric densities.

Figure 3. Comparison of our exact divergence computation with Hutchinson trace estimator. a) Error of divergence estimates introduced by the Hutchinson estimator increasingly scatters with growing number of particles as the variance scales with $O(N \cdot D)$. b) Free-energy profile of the DW-2 potential and importance-weighed estimates from a trained symmetric BG. Noisy estimates render reweighing to the exact target density impossible. c) Brute-force computation per evaluation of $v(x(t), t)$ quickly becomes prohibitively slow, while evaluation of the exact trace has constant costs.

7.2. DTO vs. OTD optimization

In this experiment we show that by simply regularizing $W$, e.g. using weight decay, OTD and DTO based optimization of the flow barely shows any difference (Figure 4 a), while the former quickly results in a significant overhead per function evaluation during the integration, which makes it impractical for online computations (Figure 3 c), such as using the flow within a sampling procedure or just for training. If we use Hutchinson estimation, the error grows quickly with the number of particles (Figure 3 a) and renders reweighing, even for the very simple DW-2 system, impossible (Figure 3 b). By having access to an exact closed-form trace, we obtain the best of both worlds: fast computation and the possibility for exact reweighing (Figure 3 b+c).

7.3. Statistical efficiency for density estimation

We compare our proposed equivariant flows to a non-equivariant flow where $v(x(t), t)$ is given by a simple fully-connected neural network. As brute-force computation of the divergence quickly becomes prohibitively slow for the LJ-13 system, we rely on Hutchinson-estimation during training and compute the exact divergence only during evaluation.

We generate a training data set by sampling $10/100/1,000/10,000$ samples from a long MCMC trajectory (throwing away 1,000 burn-in samples to enforce equilibration). After training we evaluate the likelihood of the model on an independent 10,000 trajectory.
We train both flows using Adam with weight decay (Kingma & Ba, 2014; Loshchilov & Hutter) until convergence. For the non-symmetric flow we tested both: data augmentation by applying random rotations and permutations, and no data augmentation.

Our results show that a symmetric flow generalizes well to the unseen trajectory even in the low data regime. When applying data augmentation, the non-symmetric flow significantly performs worse (DW-4) or even fails to fit the data at all and remains close to the prior distribution (LJ-13). Without data augmentation, even using strong regularization, we observe strong over-fitting behavior: the DW-4 system can only be fitted if trained close to the full equilibrium distribution, the LJ system cannot be fitted sufficiently at all (Figure 5).

At this point it is worth to remark that the symmetric flow only requires 620 trainable parameters in order to achieve this result compared to the 5256 (DW-4) / 21671 (LJ-13) parameters of the black-box model.

Due to the simple structure of the proposed we can visualize the learned dynamics of the equivariant flow, by plotting $W$ after training (Figure 6).

7.4. Discovery of new meta-stable states

In a final experiment, we evaluate to which extend these models help discovering new meta-stable states, which have not been observed in the training data set. Here we characterize metastable states as the set of configurations $x$ that minimize to the same local minimum on the energy surface. Finding new meta-stable states is especially non-trivial for LJ systems with many particles.

Counting distinct meta-stable states Let $\psi$ be the function mapping a state $x$ onto its next meta-stable state $\psi(x)$. We implement it by minimizing $x$ w.r.t. $u(x)$ using a non-momentum optimizer until convergence and filtering out saddle-points. Then we equate two minima $\psi(x) \sim \psi(x')$, whenever they are identical up to rotations and permutations.

To avoid computing the orthogonal Procrustes problem between all minimized structures, we compute the all-distance matrix $M_d(\psi(x))$ of each minimum state, sort it in ascending order to obtain $M_{d,\text{sorted}}(\psi(x))$ and equate two structures $\psi(x) \sim_{\text{approx}} \psi(x')$, whenever

$$\|M_{d,\text{sorted}}(x) - M_{d,\text{sorted}}(x')\| < \epsilon,$$

where $\epsilon \ll 1$ is a threshold depending on the system. This ensures that $\psi(x) \sim \psi(x') \implies \psi(x) \sim_{\text{approx}} \psi(x')$, however the inverse direction might not hold. Thus, the reported numbers on the count of unique local minima found remain a lower bound.

**DW-4** For this system, we can fully enumerate those five meta-stable minima between which the system jumps in equilibrium.

We train both a symmetric flow and a non-symmetric flow on a single minimum state perturbed by a tiny amount of Gaussian noise until convergence. Then we sample 10,000 structures from both models and compute the set of unique minima.

While the non-symmetric flow model can only reproduce the minimum state it has been trained on, the symmetric flow discovers all minimum states of the system (see Figure 7 a).

**Table 1.** Count of unique minima states discovered: displayed are means and standard deviation over 10 independent rounds.

| Method        | $(-70, -60)$ | $(-80, -70)$ | $(-\infty, -80)$ |
|---------------|--------------|--------------|------------------|
| TRAINING      | 0            | 3            | 0                |
| MCMC          | 0.30 ± 0.46  | 3.80 ± 1.47  | 0.60 ± 0.49      |
| MCMC-LONG     | 7.67 ± 2.05  | 25.0 ± 2.94  | 1.00 ± 0.00      |
| EQ-FLOW       | 5.4 ± 1.85   | 15.50 ± 2.94 | 1.00 ± 0.00      |

**LJ-13** Finding meta-stable minima with low energies is a much more challenging task for the LJ system. Here we compare the proposed symmetric flow model to standard sampling by (1) training on a short equilibrium MCMC trajectory consisting of 1,000 samples, (2) sampling 1,000 samples from the generator distribution after training, and (3) counting the amount of unique minima states found according to the procedure described above. The amount of unique minima found is compared to sampling an independent equilibrium MCMC trajectory having the same amount of samples as the training set and a second long trajectory with 20,000 samples.

As can be seen from Table 1 the symmetric flow model clearly outperforms naive sampling in finding low-energy
Equivariant Flows: exact likelihood generative learning for symmetric densities.

Figure 7. a) On the left: Minimum state used for training in the DW-4 system. Upper rows: samples from equivariant flow (blue) and corresponding minimum states (red). Bottom rows: samples from non-equivariant flow (blue) and corresponding minimum states (red). b) Exemplary unique minima states from the LJ-13 system generated within the three given energy intervals. The top state marks the global minimum, which consists of a perfect icosahedron with one particle in the center.

8. Discussion

We presented a construction principle to incorporate symmetries of densities defined over $\mathbb{R}^n$ into the structure of normalizing flows. We further demonstrated the superior generalization capabilities of such symmetry-preserving flows compared to non-symmetry-preserving ones on two physics-motivated particle systems, which are difficult to sample with classic methods. Our proposed equivariant kernel-based vector fields have several structural advantages over black box CNFs, such as an analytically computable divergence, explicit handling of numerical stability and very few parameters.

We conclude this paper with two points which remain open for future research:

- Is the class of $G$-equivariant flows starting from a Gaussian prior density $\rho$ sufficient to model any $G$-invariant target density $\rho'$ (assuming $\rho' > 0$ everywhere)?

  Possible extension of this class could involve $G$-steerable flows, or a formalism to model the density transport directly in the quotient space.

- How do we scale equivariant flows to large particle systems?

  Here we foresee combining our results either with symplectic integrators or by restricting the class of dynamics functions to those which are numerically stable e.g. as done in (Manek & Kolter, 2020).

References

Albergo, M. S., Kanwar, G., and Shanahan, P. E. Flow-based generative models for markov chain monte carlo in lattice field theory. *Phys. Rev. D*, 100:034515, 2019.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

Benamou, J.-D. and Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Bennett, C. H. Efficient estimation of free energy differences from monte carlo data. *J. Chem. Phys.*, 22:245–268, 1976.

Berg, R. v. d., Hasenclever, L., Tomczak, J. M., and Welling, M. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

Chen, C., Li, C., Chen, L., Wang, W., Pu, Y., and Carin, L. Continuous-time flows for efficient inference and density estimation. *arXiv preprint arXiv:1709.01179*, 2017.

Chen, T. Q. and Duvenaud, D. K. Neural networks with cheap differential operators. In *Advances in Neural Information Processing Systems*, pp. 9961–9971, 2019.
Equivariant Flows: exact likelihood generative learning for symmetric densities.

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.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In Advances in Neural Information Processing Systems, pp. 9913–9923, 2019.

De Cao, N., Titov, I., and Aziz, W. Block neural autoregressive flow. arXiv preprint arXiv:1904.04676, 2019.

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. arXiv preprint arXiv:1605.08803, 2016.

Dormand, J. R. and Prince, P. J. A family of embedded runge-kutta formulae. Journal of computational and applied mathematics, 6(1):19–26, 1980.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In Advances in Neural Information Processing Systems, pp. 7509–7520, 2019.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. MADE: Masked autoencoder for distribution estimation. In International Conference on Machine Learning, pp. 881–889, 2015.

Gholami, A., Keutzer, K., and Biros, G. Anode: Unconditionally accurate memory-efficient gradients for neural odes. arXiv preprint arXiv:1902.10298, 2019.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Fjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367, 2018.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. arXiv preprint arXiv:1804.00779, 2018.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. 1989.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In Advances in Neural Information Processing Systems, pp. 10215–10224, 2018.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. arXiv preprint arXiv:1910.00753, 2019.

Laio, A. and Parrinello, M. Escaping free energy minima. Proc. Natl. Acad. Sci. USA, 99:12562–12566, 2002.

Li, S.-H. and Wang, L. Neural network renormalization group. Phys. Rev. Lett., 121:260601, 2018.

Liu, J., Kumar, A., Ba, J., Kiros, J., and Swersky, K. Graph normalizing flows. In Advances in Neural Information Processing Systems, pp. 13556–13566, 2019.

Liu, X., Qi, C. R., and Guibas, L. J. Flownet3d: Learning scene flow in 3d point clouds. arXiv:1806.01411, 2019b.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. arXiv:1711.05101. In Proceedings of the ICLR 2018 Conference Blind Submission, Vancouver, BC, Canada, volume 30.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101, 2017.

Louizos, C. and Welling, M. Multiplicative normalizing flows for variational bayesian neural networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2218–2227. JMLR. org, 2017.

Manek, G. and Kolter, J. Z. Learning stable deep dynamics models. arXiv preprint arXiv:2001.06116, 2020.

McCann, R. J. Polar factorization of maps on riemannian manifolds. Geometric & Functional Analysis GAFA, 11 (3):589–608, 2001.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural importance sampling. arXiv preprint arXiv:1808.03856, 2018.

Nicolò, K. A., Nakajima, S., Strodthoff, N., and Chiew, Samek, W., Müller, K.-R., and Kessell, P. Asymptotically unbiased estimation of physical observables with neural samplers. arXiv:1910.13496, 2019.

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.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. Parallel wavenet: Fast high-fidelity speech synthesis. arXiv preprint arXiv:1711.10433, 2017.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In Advances in Neural Information Processing Systems, pp. 2338–2347, 2017.
Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, D. J., Racanière, S., Higgins, I., and Toth, P. Equivariant hamiltonian flows. *arXiv preprint arXiv:1909.13739*, 2019.

Tabak, E. G. and Turner, C. V. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

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.

Tomczak, J. M. and Welling, M. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

Torrie, G. M. and Valleau, J. P. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *J. Comp. Phys.*, 23:187–199, 1977.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

Wales, D. J. and Doye, J. P. K. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 100 atoms. *J. Phys. Chem. A*, 101:5111–5116, 1997.

Wu, H., Paul, F., Wehmeyer, C., and Noé, F. Multiensemble markov models of molecular thermodynamics and kinetics. *Proc. Natl. Acad. Sci. USA*, 113:E3221–E3230, 2016. doi: 10.1073/pnas.1525092113.

Zhang, L., Wang, L., et al. Monge-ampere flow for generative modeling. *arXiv preprint arXiv:1809.10188*, 2018.
Supplementary Material

A. Proofs

A.1. Proof of thm. 1

Let $V = \mathbb{R}^n$ and $\rho: V \to \mathbb{R}_{\geq 0}$ be a probability density on $V$. Let $G$ be a group acting on $V$ and let $R: G \to GL(n), g \to R_g$ be a representation of $G$ over $V$. As $V$ is finite-dimensional every $R_g$ is represented by a matrix and thus $\det R_g$ is well-defined. Furthermore, for a function $f \in C^1(\mathbb{R}^n, \mathbb{R}^m)$ let $J_f(x) \in \mathbb{R}^{n \times m}$ denote its Jacobian evaluated at $x$ and define the push-forward density of $\rho$ along a diffeomorphism $f \in C^1(V, V)$ by

$$\rho_f(x) := \rho(f^{-1}(x)) |\det J_f(x)|.$$

**Lemma 1.** Let $A \in GL(n)$, if $\rho(Ax) = \rho(x)$ for all $x \in V$, then $\det A \in \{-1, 1\}$

**Proof.** Set $a: V \to V, x \mapsto Ax$. By substituting $y = a^{-1}x$ we get

$$1 = \int_V \rho(x) dx = \int_{a^{-1}(V)} \rho(a(y)) |\det A| dy = \int_V \rho(y) |\det A| dy = |\det A| \int_V \rho(z) dy = |\det A|$$

Let $G > H$ and $h \in H$. From Lemma 1 we get $\det R_h \in \{-1, 1\}$ for each $h \in H$. Define the transformation $T_h: V \to V, x \mapsto R_h x$. If $f \in C^1(V, V)$ is $H$-equivariant, it means $f \circ T_h = T_h \circ f$ for each $h \in H$. If $\rho$ is an $G$-invariant density it means $\rho \circ T_g = \rho$. Together with the lemma we obtain

$$\rho_f(R_h x) = \rho_f(T_h(x)) = \rho_f(T_h(x)) \left| \det J_{T_h}(x) \right| = \rho_{T_h} \circ f(x) = \rho_{f \circ T_h}(x) = \rho((T_h \circ f^{-1})(x)) \left| \det J_{T_h \circ f^{-1}}(x) \right| = (\rho \circ T_h \circ f^{-1})(x) \left| \det J_{T_h}(f^{-1}(x)) J_{f^{-1}}(x) \right| = (\rho \circ f^{-1})(x) \left| \det J_{T_h}(f^{-1}(x)) \right| \left| \det J_{f^{-1}}(x) \right| = \rho(f^{-1}(x)) \left| \det J_{f^{-1}}(x) \right| = \rho(f^{-1}(x)) \left| \det J_{f^{-1}}(x) \right| = \rho_f(x)$$
A.2. Proof of thm. 2

Proof. Let $h \in H$ and $R_h$ be its representation. Let $v$ be an $H$-equivariant vector field. Then

$$F_{v,T}(R_hz) = R_h x_{v,z}(0) + \int_0^T dt \, v(R_h x_{v,z}(t),t)$$

$$= R_h x_{v,z}(0) + \int_0^T dt \, R_h v(x_{v,z}(t),t)$$

$$= R_h \left( x_{v,z}(0) + \int_0^T dt \, v(x_{v,z}(t),t) \right).$$

This implies that the bijection $F_{v,T}$ for each $T \in [0, \infty)$ given by solving

$$x_{v,z}(0) = z$$

$$\frac{d}{dt} x(t) = v(x_{v,z}(t),t)$$

is $H$-equivariant.

A.3. Invariant prior density

Subtracting the CoM of a system $x \in \mathbb{R}^{N \times D}$ and obtaining a CoM-free $\tilde{x}$, can be considered a linear transformation

$$\tilde{x} = Ax$$

with

$$A = I_D \otimes \left( I_N - \frac{1}{N} 1_N 1_N^T \right)$$

where $I_k$ is the $k \times k$ identity matrix and $1_k$ the $k$-dimensional vector containing all ones.

$A$ is a symmetric projection operator, i.e. $A^2 = A$ and $A^T = A$. Furthermore $\text{rank}[A] = (N - 1)D$. Finally, we have $Ay = y$ for each $y \in U$.

If we equip $\mathbb{R}^n$ with an isotropic density $\rho = \mathcal{N}(0, I_n)$, this implies the subspace density $\tilde{\rho} = \mathcal{N}(0, AI_n A^T) = \mathcal{N}(0, AA^T)$. Thus, sampling from $\rho$ and projecting by $A$ achieves sampling from $\tilde{\rho}$ trivially. On the other hand, if we have $y \in U$, then $\|y\|_2^2 = \|Ay\|_2^2$ and thus $\rho(y) = \tilde{\rho}(y)$.

B. Technical details

In this section we show the hyperparameters and optimization details used for the experiments presented in this work. For all experiments we used ML-training to train the models on the given training data ($\lambda = 0$). The only exception is the discovery of new meta-stable states (7.4) for the DW-4 system, where we used a combination of ML-training and KL-training with $\lambda = 0.5$ after pretraining both models with ML-training.

B.1. Equivariant kernel flow

For the DW-4 system we fixed 50, 10 kernel means $\mu_{K,l}$, $\mu_{R,l}$ equispaced in $[0, 8]$, $[0, 1]$ for distances and times respectively. The bandwidths $\gamma_{K,l}, \gamma_{R,l}$ of the kernels have been initialized with $0.3, 0.5$ and were optimized during the training process. The total model ended up having 620 trainable parameters.

For the LJ-13 system we fixed 50 kernel means $\mu_{K,l}$ in $[0, 16]$ concentrated around $r_m = 1$ with increasing distance to each other towards the interval bounds. Similarly bandwidths $\gamma_{K,l}$ are initialized narrowly close to $r_m$ increasing towards the interval bounds. We placed the 10 kernels $\mu_{R,l}$ for the time-dependent component equispaced in $[0, 1]$. The bandwidths $\gamma_{R,l}$ where initialized with narrower bandwidths around $t = 0.5$ and smearing out the closer they reach the interval boundaries. Again bandwidths were optimized during the training process. This resulted in a total of 620 trainable parameters.

As regularization is important to efficiently train our architecture using fixed step-size solvers our models were optimized using AdamW, a modified implementation with fixed weight-decay (Loshchilov & Hutter, 2017) using a learning rate of 0.005, weight decay of 0.01 and a batch size of 64 samples until convergence. The number of bandwidths were optimized during the training process. This resulted in a total of 620 trainable parameters.
B.2. Non equivariant nODE flow

For the DW-4 system we used a dense neural network with layer sizes $[64, 64]$ and $\tanh$ activation functions. This resulted in a total of 5256 trainable parameters.

For the LJ-13 system we used a dense neural network with layer sizes $[64, 128, 64]$ and $\tanh$ activation functions. This resulted in a total of 21671 trainable parameters. For the optimization we used Adam with a learning rate of 0.005. We optimized the model with a batch size of 64 samples until convergence.

B.3. MCMC trajectories

For each system, a training and a test trajectory were obtained with Metropolis Monte-Carlo, where we optimized the width of the Gaussian proposal density by maximizing $\alpha \cdot s$, with $\alpha$ being the acceptance rate computed from short trajectories and $s$ the Gaussian standard deviation (step size). The optimal step sizes are $s = 0.5$ for the DW-4 system and $s = 0.025$ for the LJ-13 system. To ensure that all samples steam from the equilibrium distribution we discard a large number of initial samples. For the DW-4 system the initial 1000 samples are discarded, while we discard 20000 for the LJ-13 system.

B.4. Benchmark systems

Throughout all experiments we chose the same parameters for our two benchmark systems.

For the DW-2/DW-4 system we chose $a = 0, b = -4, c = 0.9, d_0 = 4$ and a dimensionless temperature factor of $\tau = 1$.

For the LJ-13 system we chose $r_m = 1, \epsilon = 2$ and a dimensionless temperature factor of $\tau = 1$.

B.5. Error bars

Error bars in all plots are given by one standard deviation.

In Figure 3 a) we show errors for 1000 estimations per particle count. In Figure 3 b) errors are displayed for 100 reweighed bootstrapped sub-samples. In Figure 3 c) time was measured for 100 estimations per particle count per method.

In Figure 4 a) we show 3 runs per method.

In Figure 5 we show 5 runs per model/system/training set size.

B.6. Computing infrastructure

All experiments were conducted on a GeForce GTX 1080 Ti with 12 GB RAM.