Equivariant Manifold Flows

Anonymous Author(s)
Affiliation
Address
email

Abstract

Tractably modelling distributions over manifolds has long been an important goal in the natural sciences. Recent work has focused on developing general machine learning models to learn such distributions. Though these are significant advances, many problems require the learned distribution to obey symmetries inherent to the manifold. Up until now, the theory of invariant learning on manifolds was underdeveloped—models were unable to leverage these symmetries to aid learning. In this paper, we lay the theoretical foundations for learning symmetry-invariant distributions on arbitrary manifolds via equivariant manifold flows. We demonstrate the efficacy of our approach in empirical settings, including the learning of gauge invariant densities over $SU(n)$ in the context of quantum field theory.

1 Introduction

Learning probabilistic models for data has long been the focus of many problems in machine learning and statistics. Though much effort has gone into learning models over Euclidean space [4, 10, 11], less attention has been allocated to learning models over non-Euclidean space, despite the fact that this is still a very important problem. Notably, density learning over non-Euclidean spaces has applications ranging from quantum field theory in physics [27] to motion estimation in robotics [9] to protein-structure prediction in computational biology [12].

Continuous normalizing flows [4, 11] are powerful generative models for learning structure in complex data due to their tractability and theoretical guarantees. Recent work [18, 19] has extended the framework of continuous normalizing flows to the setting of distribution learning on Riemannian manifolds, enabling applications in the aforementioned contexts.

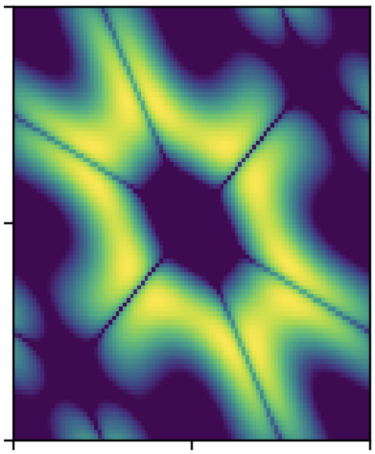


Figure 1: An example of a density on $SU(3)$ that is invariant to conjugation by $SU(3)$. The x-axis and y-axis are the angles θ_1 and θ_2 for eigenvalues $e^{i\theta_1}$ and $e^{i\theta_2}$ of a matrix in $SU(3)$. The axis range is $-\pi$ to π.

Although tractable and principled learning of continuous normalizing flows was a considerable step forward, this is insufficient for many problems in the natural sciences. Several applications, for example sampling coupled particle systems in physical chemistry [15] or sampling for $SU(n)$[1] lattice gauge theories in theoretical physics [2], require distributions to have symmetries that are nontrivial to enforce. More precisely, these symmetries are invariances with respect to action by an isometry subgroup of the underlying manifold. For example, one could imagine learning a density on the sphere that is invariant to rotation around an axis; this is an example of an isometry subgroup

[1] $SU(n)$ denotes the special unitary group $SU(n) = \{X \in \mathbb{C}^{n\times n} \mid X^*X = I,\ \det(X) = 1\}$.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

invariant[2] density. For a less trivial example, note that when learning a flow-based sampler for $SU(n)$ in the context of lattice QFT [2], the learned density must be invariant to conjugation by $SU(n)$ (see Figure 1 for a density on $SU(3)$ that exhibits the requisite symmetry).

One might ask, could an invariant density be learned simply by taking the quotient of the underlying manifold by the symmetry group and learning over the result with a general manifold density learning method such as NMODE [18]? Though this seems plausible, it is problematic for several reasons:

1. First, this approach requires charts for the quotient manifold, explicit representations of its tangent spaces, and formulas for the exponential map and other manifold operations. The nontrivial structure of many quotient manifolds can make it hard to get such a representation.
2. Second, the quotient manifold often has a boundary. For instance, in the simple case mentioned above of the sphere invariant to rotations about an axis, the quotient manifold is a closed interval (representing possible latitudes on a globe). Learning a continuous normalizing flow over such a manifold-with-boundary is not principled (and would in practice be very numerically unstable) since probability mass can "flow out" on the boundary.
3. Third, even if the quotient is a manifold without boundary for which we have a clear characterization, the quotient may have a discrete structure that induces artifacts in the invariant density in the process of learning. In particular, this is the case in Boyda et al. [2]: although the quotient is a manifold without boundary, the flow construction over the quotient induces unnatural artifacts in the density.

Motivated by the above drawbacks, our goal is to design a manifold continuous normalizing flow that learns over the original manifold while maintaining the requisite symmetry invariance. Note that a typical manifold CNF [18] does not preserve invariance. How does one ensure the flow produces an invariant distribution? To ensure invariance, we will show that it suffices for the flow vector field to be *equivariant* with respect to the symmetry. Modelling equivariant vector fields on manifolds is not trivial and has not been done before in a general setting; this is one of the central contributions of our paper. In particular, when applied to the densities in Boyda et al. [2], we show we are able to avoid the unnatural artifacts of the quotient construction by learning in a continuous manner over the original manifold. A summary of our paper's contributions is as follows:

- We present a general framework for learning equivariant manifold flows: in our setup, the flows can be learned over arbitrary Riemannian manifolds while explicitly incorporating symmetries inherent to the problem.
- We demonstrate the efficacy of our approach by learning gauge invariant densities over $SU(n)$ in the context of quantum field theory. Our approach learns the target densities while adhering more naturally to the target geometry in comparison to the prior approach.
- We highlight the benefit of incorporating symmetries into manifold flow models by comparing directly against previous general manifold density learning approaches. We show that when the model is not aware of symmetries inherent to the problem, the learned density is of considerably worse quality and, in particular, violates said symmetries.

2 Related Work

Our work aims to build directly on pre-existing normalizing flow models that learn densities over manifolds, and to enable these models to leverage symmetries inherent to the densities over the underlying manifold. In this section we cover the related work on which we build; in particular, the two facets of this body of work have to do with learning normalizing flows on manifolds as well as with equivariant machine learning broadly. Our work combines the two and enables learning normalizing flows in the most general setting: on arbitrary Riemannian manifolds and with specified symmetries taken into account.

Normalizing Flows on Manifolds Normalizing flows on Euclidean space have long been touted as powerful generative models that work by learning to map samples from an intractable density to a tractable, easy to sample from prior in an invertible manner, so as to enable sampling from the

[2]This specific isometry subgroup is known as the isotropy group at a point of the sphere intersecting the axis.

complex distribution [4, 11]. In contrast to other generative models such as GANs [10], normalizing flows learn to sample the real target distribution, since they account for volume changes induced by invertible transformations that map to the prior. In a rather concrete sense, this makes them theoretically principled. As such, they are ideal candidates for generalization beyond the Euclidean setting, where a careful, theoretically principled modelling approach is necessary.

Hence, many have attempted such a generalization, either through manifold-specific or general constructions. The paper [25] introduced constructions specific to tori and spheres, while [1] introduced constructions for hyperbolic space. Following this work, [8, 18, 19] concurrently introduced a general construction by extending Neural ODEs [4] to the setting of Riemannian manifolds. Our work takes inspiration from the methods of [18, 19] and generalizes them further to enable learning that takes into account symmetries of the target density.

Equivariant Machine Learning Equivariance has been recently discussed in the context of machine learning [5–7, 16] and is broadly useful for learning models that are able to leverage symmetries inherent to specific applications. Köhler et al. [15], in particular, uses equivariant normalizing flows to enable learning symmetric densities over Euclidean space. The authors note their approach is better suited to density learning in some physical chemistry settings (when compared to general purpose normalizing flows), since they take into account the symmetries of the problem.

Equivariance is important for learning over manifolds for a similar reason. Building models that leverage symmetries inherent to the relevant problem is not only important for better modelling[3], but the symmetry is frequently a strict requirement of the problem, i.e. the learned density must obey the symmetry. To give a particular example, Boyda et al. [2] introduced equivariant flows for $SU(n)$ via a manifold-specific construction; the densities learned were required to be invariant to conjugation by $SU(n)$, a necessary condition for applying said densities to create a flow-based sampler for $SU(n)$ in the context of lattice quantum field theory.

Not much other work has been done for learning such invariant distributions over manifolds. Note that Boyda et al. [2] produces a method relevant only to learning a density over $SU(n)$ invariant to conjugation by $SU(n)$. In contrast, our paper presents a fully general way to learn flows that induce symmetry invariant distributions, i.e. we enable learning a density on a manifold invariant to action by any isometry subgroup. Moreover, we show that we can learn the densities in Boyda et al. [2] while more naturally adhering to the desired geometry, since we avoid the artifacts induced by the discrete construction in Boyda et al. [2].

3 Background

In this section, we provide a terse overview of necessary concepts for understanding our paper. In particular, we address fundamental notions from Riemannian geometry as well as the basic set-up of normalizing flows on manifolds. For a more detailed introduction to Riemannian geometry, we refer the reader to a text such as Kobyzev et al. [14], Lee [17].

3.1 Riemannian Geometry

A Riemannian manifold $(\mathcal{M}, h)$ is an n-dimensional manifold with a smooth collection of inner products $(h_x)_{x\in\mathcal{M}}$ for every tangent space $T_x\mathcal{M}$. The Riemannian metric h induces a distance d_h on the manifold.

A diffeomorphism $f : \mathcal{M} \to \mathcal{M}$ is called an isometry if $h(D_x f(u), D_x f(v)) = h(u, v)$ for all tangent vectors $u, v \in T_x\mathcal{M}$ where $D_x f$ is the differential of f. Note that isometries preserve the manifold distance function. The collection of all isometries forms a group G, which we call the isometry group of the manifold $\mathcal{M}$.

Riemannian metrics also allow for a natural analogue of gradients on $\mathbb{R}^n$. For a function $f : \mathcal{M} \to \mathbb{R}$, we define the Riemannian gradient $\nabla_x f$ to be the vector on $T_x\mathcal{M}$ such that $h(\nabla_x f, v) = D_x f(v)$ for $v \in T_x\mathcal{M}$.

[3] For example, asteroid impacts on the sphere can be modelled as being approximately invariant to rotation about the Earth's axis.

3.2 Normalizing Flows on Manifolds

Let $(\mathcal{M}, h)$ be a Riemannian manifold. A normalizing flow on $\mathcal{M}$ is a diffeomorphism $f_\theta : \mathcal{M} \to \mathcal{M}$ (parametrized by θ) that transforms a prior density ρ to model density ρ_{f_θ}. The model distribution can be computed via the change of variables equation:

$$\rho_{f_\theta}(x) = \rho\left(f_\theta^{-1}(x)\right)\left|\det \frac{\partial f_\theta^{-1}(x)}{dx}\right| = \rho\left(f_\theta^{-1}(x)\right)\left|\det J_{f_\theta^{-1}}(x)\right|.$$

3.3 Equivariance and Invariance

Let G be an isometry subgroup of $\mathcal{M}$. We notate the action of an element $g \in G$ on $\mathcal{M}$ by the function $R_g : \mathcal{M} \to \mathcal{M}$.

Equivariant and Invariant Functions We say that a function $f : \mathcal{M} \to \mathcal{N}$ is equivariant if, for symmetries $g_x : \mathcal{M} \to \mathcal{M}$ and $g_y : \mathcal{N} \to \mathcal{N}$, $f \circ g_x = g_y \circ f$. We say a function $f : \mathcal{M} \to \mathcal{N}$ is invariant if $f \circ g_x = f$. When X and Y are manifolds, the symmetries g_x and g_y are isometries.

Equivariant Vector Fields Let $X : \mathcal{M} \times [0, \infty) \to T\mathcal{M}$, $X(m, t) \in T_m\mathcal{M}$ be a time-dependent vector field on manifold $\mathcal{M}$, with base point $x_0 \in \mathcal{M}$.

X is a G-equivariant vector field if $\forall (m, t) \in \mathcal{M} \times [0, \infty)$, $X(R_g m, t) = (D_m R_g)X(m, t)$.

Equivariant Flows A flow f on a manifold $\mathcal{M}$ is G-equivariant if it commutes with actions from G, i.e. we have $R_g \circ f = f \circ R_g$.

Invariance of Density For a group G, a density ρ on a manifold $\mathcal{M}$ is G-invariant if, for all $g \in G$ and $x \in \mathcal{M}$, $\rho(R_g x) = \rho(x)$, where R_g is the action of g on x.

4 Invariant Densities from Equivariant Flows

Our goal in this section is to describe a tractable way to learn a density over a manifold that obeys a symmetry given by an isometry subgroup G. Parameterizing a density that obeys this symmetry directly is highly nontrivial. Hence we will prove the following set of implications that will yield a tractable solution to this problem:

1. **G-invariant potential $\Rightarrow$ G-equivariant vector field (Theorem 1).** We show that given a G-invariant potential function $f : \mathcal{M} \to \mathbb{R}$, the vector field ∇f is G-equivariant.
2. **G-equivariant vector field $\Rightarrow$ G-equivariant flow (Theorem 2).** We show that a G-equivariant vector field on $\mathcal{M}$ uniquely induces a G-equivariant flow.
3. **G-equivariant flow $\Rightarrow$ G-invariant density (Theorem 3).** We show that given a G-invariant prior distribution and a G-equivariant flow, the output flow density is G-invariant.

We claim that constructing a G-invariant potential function on a manifold is far simpler than directly parameterizing a G-invariant density or a G-equivariant flow. We shall give explicit examples of G-invariant potential constructions in Section 6 that induce a desired density invariance. Hence we see, taking into account the assumptions of Theorem 3, that if we have a prior distribution on the manifold that obeys the requisite invariance, we can use a G-invariant potential to produce a flow that allows us to use the continuous normalizing flow framework to learn an output density with the desired invariance.

Moreover, we show in Theorem 4 that considering equivariant flows generated from invariant potential functions suffices to learn any smooth distribution over a closed manifold, as measured by Kullback-Leibler divergence.

We defer the proofs of all theorems to the appendix.

4.1 Equivariant Gradient of Potential Function

We start by showing how to construct G-equivariant vector fields from G-invariant potential functions.

To design an equivariant vector field X, it is sufficient to set the vector field dynamics of X as the gradient of some G-invariant potential function $\Phi : \mathcal{M} \to \mathbb{R}$:

Theorem 1. *Let $(\mathcal{M}, h)$ be a Riemannian manifold and G be its group of isometries (or an isometry subgroup). If $\Phi : \mathcal{M} \to \mathbb{R}$ is a smooth G-invariant function, then the following diagram commutes for any $g \in G$:*

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{R_g} & \mathcal{M} \\ \downarrow{\scriptstyle \nabla\Phi} & & \downarrow{\scriptstyle \nabla\Phi} \\ T\mathcal{M} & \xrightarrow{DR_g} & T\mathcal{M} \end{array}$$

or $\nabla_{R_g u}\Phi = D_u R_g(\nabla_u \Phi)$. Hence $\nabla\Phi$ is a G-equivariant vector field. This condition is also tight in the sense that it only occurs if G is the group of isometries.

Hence, as long as one can construct a G-invariant potential function, one can obtain the desired equivariant vector field. By this construction, a parameterization of G-invariant potential functions yields a parameterization of (some) G-equivariant vector fields.

4.2 Constructing Equivariant Manifold Flows from Equivariant Vector Fields

To construct equivariant manifold flows, we will use tools from manifold ordinary differential equations (ODEs) and continuous normalizing flows (CNFs). In particular, note the definition below:

Manifold Continuous Normalizing Flows A manifold continuous normalizing flow with base point z is a function $\gamma : [0, \infty) \to \mathcal{M}$ that satisfies the manifold ODE

$$\frac{d\gamma(t)}{dt} = X(\gamma(t), t)\ , \gamma(0) = z.$$

We define $F_{X,T} : \mathcal{M} \to \mathcal{M}$, $z \mapsto F_{X,T}(z)$ to map any base point $z \in \mathcal{M}$ to the value of the CNF starting at z, evaluated at time T. This function is known as the (vector field) flow of X.

Observe that there exists a natural correspondence between equivariant flows and equivariant vector fields, by the following theorem:

Theorem 2. *Let $(\mathcal{M}, h)$ be a Riemannian manifold, and G be its isometry group (or one of its subgroups). Let X be any time-dependent vector field on $\mathcal{M}$, and $F_{X,T}$ be the flow of X. Then X is a G-equivariant vector field if and only if $F_{X,T}$ is a G-equivariant flow.*

Hence we can obtain an equivariant flow from an equivariant vector field, and vice versa.

4.3 Invariant Manifold Densities from Equivariant Flows

We now show that G-equivariant flows induce G-invariant densities. Note that we require the group G to be an isometry subgroup in order to control the density of ρ_f, and the following theorem does not hold for general diffeomorphism groups.

Theorem 3. *Let $(\mathcal{M}, h)$ be a Riemannian manifold, and G be its isometry group (or one of its subgroups). If ρ is a G-invariant density on $\mathcal{M}$, and f is a G-equivariant diffeomorphism, then $\rho_f(x)$ is also G-invariant.*

In the context of manifold normalizing flows, Theorem 3 implies that if the prior density on $\mathcal{M}$ is G-invariant and the flow is G-equivariant, the resulting output density will be G-invariant. In the context of the overall set-up, this reduces the problem of constructing a G-invariant density to the problem of constructing a G-invariant potential function.

4.4 Sufficiency of Flows Generated via Invariant Potentials

One may pose the question of whether it suffices to use equivariant flows generated from invariant potentials to learn arbitrary invariant distributions over manifolds. In particular, it is reasonable to have some concerns about limited expressivity, since it is unclear whether any equivariant flow can be generated in this way. We alleviate these concerns for our use cases by proving that equivariant flows obtained from invariant potential functions suffice to learn any smooth invariant distribution over a closed manifold, as measured by Kullback-Leibler divergence.

Theorem 4. *Let $(\mathcal{M}, h)$ be a closed Riemannian manifold. Let ρ, π be smooth G-invariant distributions over said manifold, and let $D_{KL}(\rho||\pi)$ be the Kullback–Leibler divergence between distributions ρ and π. If we choose a function $g : \mathcal{M} \to \mathbb{R}$ such that for $x \in \mathcal{M}$,*

$$g(x) = \log\left(\frac{\pi(x)}{\rho(x)}\right).$$

Then we have:

$$\frac{\partial}{\partial t} D_{KL}(\rho||\pi) = -\int_{\mathcal{M}} \rho \exp(g) \|\nabla g\|^2 \, dx \leq 0.$$

In particular, note that if the target distribution is π and the current distribution is ρ, if we set g to be $\log(\pi(x)/\rho(x))$ and g is the potential from which the flow is obtained, then the Kullback-Leibler divergence between π and ρ is monotonically decreasing by Theorem 4. This means precisely that considering flows generated by invariant potential functions is sufficient to learn any smooth invariant target distribution on a closed manifold, as measured by Kullback-Leibler divergence.

5 Learning Invariant Densities with Equivariant Flows

In this section, we discuss implementation details of the methodology given in Section 4. In particular, we describe the equivariant manifold flow model assuming an invariant potential is given, provide two examples of invariant potential constructions on different manifolds, and discuss how training is performed depending on the target task.

5.1 Equivariant Manifold Flow Model

We assume that a G-invariant potential function $f : \mathcal{M} \to \mathbb{R}$ is given. The equivariant flow model works by using automatic differentiation [23] on f to obtain ∇f, using this for the vector field, and integrating in a step-wise fashion over the manifold. Specifically, forward integration and change-in-density (divergence) computations utilize the Riemannian Continuous Normalizing Flows [19] framework. This flow model is used in tandem with a specific training procedure (described in Section 5.3) to obtain a G-invariant model distribution that approximates some target.

5.2 Constructing G-invariant Potential Functions

In this subsection, we present two constructions of invariant potentials on manifolds. Note that a symmetry of a manifold (i.e. action by an isometry subgroup) will leave part of the manifold free. The core idea of our invariant potential construction is to parameterize a neural network on the free portion of the manifold. While the two constructions we give below are certainly not exhaustive, they illustrate the versatility of our method, which is applicable to general manifolds.

5.2.1 Isotropy Invariance on S^2

Consider the sphere S^2, which is the Riemannian manifold $\{v \in \mathbb{R}^3 : \|v\| = 1\}$ with the induced pullback metric. The isotropy group for a point v is defined as the subgroup of the isometry group which fixes v, i.e. the set of rotations around an axis that passes through v. In practice, we let $v = (0, 0, 1)$, so the isotropy group is the group of rotations on the xy-plane. An isotropy invariant density would be invariant to such rotations, and hence would look like a horizontally-striped density on the sphere.

Invariant Potential Parameterization We design an invariant potential by applying a neural network to the free parameter. In the case of our specific isotropy group listed above, the free parameter is the z-coordinate. The invariant potential is simply a 1-input neural network on the z-coordinate of the input. As a result of this design, we see that the only variance in the learned distribution that uses this potential will be along the z-axis, as desired.

Prior Distributions For proper learning with a normalizing flow, we need a prior distribution on the sphere that respects the isotropy invariance. There are many isotropy invariant potentials on the sphere. Natural choices include the uniform density (which is invariant to all rotations) and the wrapped distribution with the center at v [21, 26]. For our experiments, we use the uniform density.

5.2.2 Conjugation Invariance on $SU(n)$

For many applications in physics (specifically gauge theory and lattice quantum field theory), one works with the Lie Group $SU(n)$ — the group of unitary matrices with determinant 1. In particular, when modelling probability distributions on $SU(n)$ for lattice QFT, the desired distribution must be invariant under conjugation by $SU(n)$ [2]. Conjugation is an isometry on $SU(n)$, so we can model probability distributions invariant under this action with our developed theory.

Invariant Potential Parameterization We must produce a G-invariant potential function $\Phi : SU(n) \to \mathbb{R}$. Note that matrix conjugation preserves eigenvalues. Thus, for a function $\Phi : SU(n) \to \mathbb{R}$ to be invariant to matrix conjugation, it has to act on the eigenvalues of $x \in SU(n)$ as a multi-set.

We can parameterize such potential functions Φ by the DeepSet network from [28]. DeepSet is a permutation invariant neural network that acts on the eigenvalues, so the mapping of $x \in SU(n)$ is $\Phi(x) = \hat{\Phi}(\{\lambda_1(x), \ldots, \lambda_n(x)\})$ for some set function $\hat{\Phi}$.

As a result of this design, we see that the only variance in the learned distribution will be amongst non-similar matrices, while all similar matrices will be assigned the same density value.

Prior Distributions For the prior distribution of the flow, we need a distribution that respects the matrix conjugation invariance. We use the Haar measure on $SU(n)$, whose volume element is given for an $x \in SU(n)$ as $\mathrm{Haar}(x) = \prod_{i<j} |\lambda_i(x) - \lambda_j(x)|^2$ [2]. We can sample from and compute the log probabilities with respect to this distribution efficiently with standard matrix computations [20].

5.3 Training Paradigms for Equivariant Manifold Flows

There are two notable ways in which we can use the model described in Section 5.1. Namely, we can use it to learn to sample from a distribution for which we have a density function, or we can use it to learn the density given a way to sample from the distribution. These training paradigms are useful in different contexts, as we will see in Section 6.

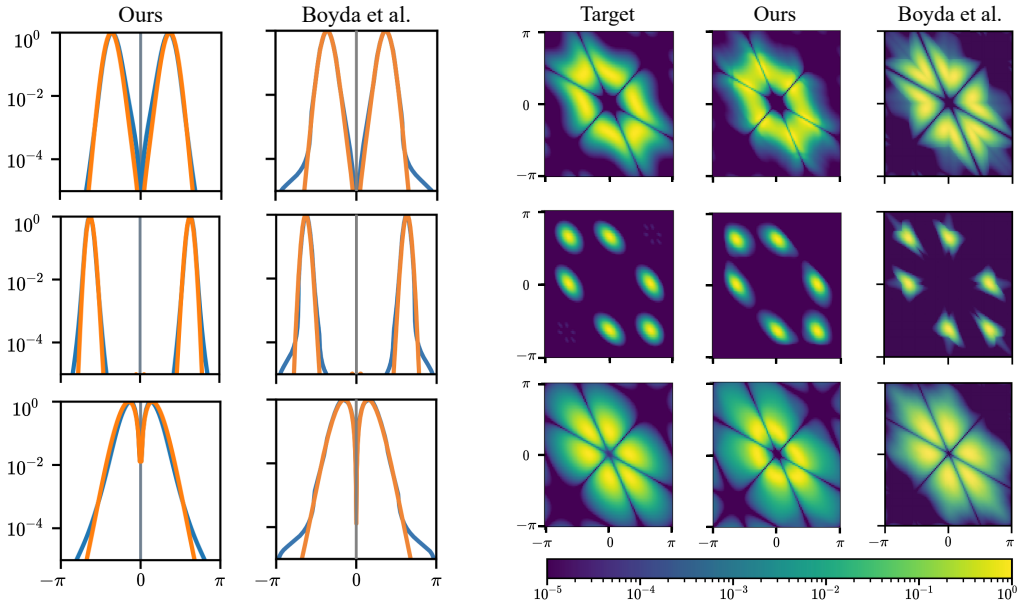
Learning to sample given an exact density. In certain settings, we are given an exact distribution and the task is to learn a tractable sampler for said distribution. For example in Boyda et al. [2], we are given conjugation-invariant densities on $SU(n)$ for which we know the exact density function (without knowledge of any normalizing constants). In contrast to procedures for normalizing flow training that use negative log-likelihood based losses, we do not have access to samples from the target distribution. Instead, we train our models by sampling from the Haar distribution on $SU(n)$, computing the Kullback-Leibler divergence between the probabilities that our model assigns to these samples and the probabilities of the target distribution evaluated at these samples, and backpropagating from this Kullback-Leibler divergence loss. When this loss is minimized, we can sample from the target distribution by sampling the prior, then forwarding the prior samples through our model. In the context of Boyda et al. [2], such a flow-based sampler is important for modelling gauge theories.

Learning the density given a sampler. In other settings, we are given a way to sample from a target distribution and want to learn the precise density for downstream tasks. For this setting, we sample the target distribution, use our flow to map it to a tractable prior, and use a negative log-likelihood-based loss. The flow will eventually learn to assign higher probabilities in sampled regions, and in doing so, will learn to approximate the target density.

6 Experiments

In this section, we utilize instantiations of equivariant manifold flows to learn densities over various manifolds of interest that are invariant to certain crucial symmetries. First, we construct flows on $SU(n)$ that are invariant to conjugation by $SU(n)$, which are important for constructing flow-based samplers for $SU(n)$ lattice gauge theories in theoretical physics [2]. In this setting, our model outperforms the ad hoc construction of Boyda et al. [2].

In addition, to demonstrate the need for enforcing equivariance of flow models, we directly compare our flow construction with a general purpose flow while learning a density with an inherent symmetry.



(a) $SU(2)$ learned densities from (Left) our model and (Right) the Boyda et al. [2] model. The target densities are in orange, while the model densities are in blue. The x-axis is θ for an eigenvalue $e^{i\theta}$ of a matrix in $SU(2)$ (note the second eigenvalue is determined as $e^{-i\theta}$). Our model has much better behavior in low-density regions, and more smoothly captures the targets in high-density regions.

(b) $SU(3)$ learned densities from (Middle) our model and (Right) the Boyda et al. [2] model for different target densities (Left). The x-axis and y-axis are the angles θ_1 and θ_2 for eigenvalues $e^{i\theta_1}$ and $e^{i\theta_2}$ of a matrix in $SU(3)$ (note the third eigenvalue is determined as $e^{-i\theta_1 - i\theta_2}$), and the probabilities correspond to colors on a logarithmic scale. Our model better captures the geometry of the target densities.

Figure 2: Comparison of learned densities on (a) $SU(2)$ and (b) $SU(3)$. All densities are normalized to have maximum value 1.

The densities we decided to use for this purpose are sphere densities that are invariant to action by the isotropy group. Our model is able to learn these densities much better than previous manifold ODE models that do not enforce equivariance of flows [18], thus showing the ability of our model to leverage the desired symmetries. In fact, even on simple isotropy-invariant densities, our model succeeds while the free model without equivariance fails.

6.1 $SU(n)$ Gauge Equivariant Neural Network Flows

Learning $SU(n)$ gauge equivariant neural network flows is important for obtaining good flow-based samplers of densities on $SU(n)$ useful for lattice quantum field theory [2]. We compare our model for $SU(n)$ gauge equivariant flows (Section 5.2.2) with that of Boyda et al. [2]. For the sake of staying true to the application area, we follow the framework of Boyda et al. [2] in learning densities on $SU(n)$ that are invariant to conjugation by $SU(n)$. In particular, our goal is to learn a flow to model a target distribution so that we may efficiently sample from it.

As mentioned above in Section 5.3, this setting follows the first paradigm in which we are given exact density functions and learn how to sample.

For the actual architecture of our equivariant manifold flows, we parameterize our potentials as DeepSet networks on eigenvalues as detailed in Section 5.2.2. The prior distribution for our model is also the Haar (uniform) distribution on $SU(n)$. Further training details are given in Appendix D.1.

6.1.1 $SU(2)$

Figure 2a displays learned densities for our model and the model of Boyda et al. [2] in the case of three particular densities on $SU(2)$ described in Appendix D.2.1. While both models match the target distributions well in high-density regions, we find that our model exhibits a considerable improvement in lower-density regions, where the tails of our learned distribution decay faster. Even

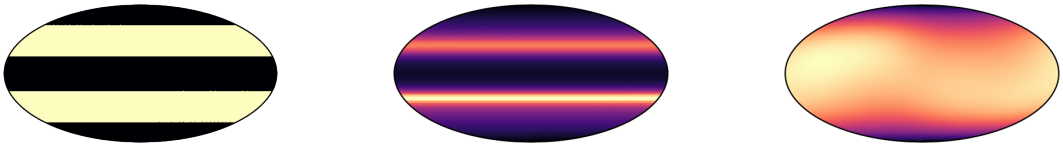
in high-density regions, our model appears to vary smoothly, with fewer unnecessary bumps and curves when compared to the densities of the model in Boyda et al. [2].

6.1.2 $SU(3)$

Figure 2b displays learned densities for our model and the model of Boyda et al. [2] in the case of three particular densities on $SU(3)$ described in Appendix D.2.2. In this case, we see that our models fit the target densities more accurately and better respect the geometry of the target distribution. Indeed, while the models of Boyda et al. [2] are often sharp and have pointed corners, our models learn densities that vary smoothly and curve in ways that are representative of the target distributions.

6.2 Modelling Invariance Matters

We also show that our equivariant condition on the manifold flow matters for efficient and accurate training when the target distribution is invariant. In particular, we again consider the sphere under the action of the isotropy group. We try to learn the isotropy invariant density given in Figure 3a and compare the results of our equivariant flow against those of a predefined manifold flow that does not explicitly model the symmetry [18]. We train for 100 epochs with a learning rate of 0.001 and a batch size of 200; our results are shown in Figure 3.



(a) Ground Truth (b) Isotropy Equivariant Flow (c) Manifold Flow

Figure 3: We compare the equivariant manifold flow and regular manifold flow on an invariant dataset. Note that our model is able to accurately capture the ground truth data distribution while the regular manifold flow struggles.

Despite our equivariant flow having fewer parameters (as both flows have the same width and the equivariant flow has an input dimension of 1), our model is able to capture the distribution much better than the base manifold flow. We believe this is due to the inductive bias of our equivariant model, which explicitly leverages the underlying symmetry.

7 Conclusion

In this work, we introduce equivariant manifold flows in a fully general context and provide the necessary theory to ensure our construction is principled. We also demonstrate the efficacy of our approach in the context of learning conjugation invariant densities over $SU(2)$ and $SU(3)$, which is an important task for sampling $SU(n)$ lattice gauge theories in quantum field theory. In particular, we show that our method can more naturally adhere to the geometry of the target densities when compared to prior work while being more generally applicable. We also demonstrate the necessity of modelling invariances by comparing against a regular manifold flow.

Further considerations. While our theory and implementations have utility in very general settings, there are still some limitations that could be addressed in future work. Further research may focus on finding other ways to generate equivariant manifold flows that do not rely on the construction of an invariant potential, and perhaps additionally on showing that such methods are sufficiently expressive to learn over open manifolds. Also, our models require a fair bit of tuning to achieve results as strong as we demonstrate in the experiments. Finally, we note that our theory and learning algorithm are too abstract for us to be sure of the future societal impacts. Still, we advance the field of deep generative models, which is known to have potential for negative impacts through malicious generation of fake images and text. Nevertheless, we do not expect this work to have negative effects in this area, as it is not clear how equivariance and manifold modelling would help.

References

[1] Joey Bose, Ariella Smofsky, Renjie Liao, Prakash Panangaden, and Will Hamilton. Latent variable modelling with hyperbolic normalizing flows. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1045–1055, 2020.

[2] Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S Albergo, Kyle Cranmer, Daniel C Hackett, and Phiala E Shanahan. Sampling using $su(n)$ gauge equivariant flows. *arXiv preprint arXiv:2008.05456*, 2020.

[3] Daniel Bump. *Lie groups*. Springer, 2004.

[4] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6571–6583, 2018.

[5] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2990–2999, 2016.

[6] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1321–1330, 2019.

[7] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[8] Luca Falorsi and Patrick Forré. Neural ordinary differential equations on manifolds. *arXiv preprint arXiv:2006.06663*, 2020.

[9] Wendelin Feiten, Muriel Lang, and Sandra Hirche. Rigid motion estimation using mixtures of projected gaussians. *Proceedings of the 16th International Conference on Information Fusion*, pages 1465–1472, 2013.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, page 2672–2680, 2014.

[11] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019.

[12] Thomas Hamelryck, John T Kent, and Anders Krogh. Sampling realistic protein conformations using local structural bias. *PLoS Computational Biology*, 2(9), 2006.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[14] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[15] Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5361–5370, 2020.

[16] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2747–2755, 2018.

[17] John M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer New York, 2013.

[18] Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 17548–17558, 2020.

[19] Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. In *Advances in Neural Information Processing Systems*, volume 33, pages 2503–2515, 2020.

[20] Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54:592–604, 2007.

[21] Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4693–4702, 2019.

[22] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[23] Adam Paszke, S. Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Neural Information Processing System Autodiff Workshop*, 2017.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[25] Danilo Jimenez Rezende, George Papamakarios, Sebastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8083–8092, 2020.

[26] Ondrej Skopek, Octavian-Eugen Ganea, and Gary B'ecigneul. Mixed-curvature variational autoencoders. *arXiv preprint arXiv:1911.08411*, 2019.

[27] Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racanière, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112, 2020.

[28] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, pages 3391–3401, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We justify our claims in theory and experiments.
   (b) Did you describe the limitations of your work? [Yes] They are described in the conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] They are described in the conclusion.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All assumptions are stated for our theorems.
   (b) Did you include complete proofs of all theoretical results? [Yes] Full proofs of all theorems are given in the Appendix.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] They are given in Section 6 and Appendix D.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not directly applicable to the types of experiments that we do in learning and visualizing densities.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] This is specific in Appendix D.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the sources of data, open-source code, and previously used models.
   (b) Did you mention the license of the assets? [Yes] The licenses of the datasets we use are mentioned in the appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]