Learning Symmetries of Classical Integrable Systems

Roberto Bondesan [1] **Austen Lamacraft** [2]

Abstract

The solution of problems in physics is often facilitated by a change of variables. In this work we present neural transformations to learn symmetries of Hamiltonian mechanical systems. Maintaining the Hamiltonian structure requires novel network architectures that parametrize symplectic transformations. We demonstrate the utility of these architectures by learning the structure of integrable models. Our work exemplifies the adaptation of neural transformations to a family constrained by more than the condition of invertibility, which we expect to be a common feature of applications of these methods.

1. Problem statement

Symmetries play a paramount role in nature and are foundational aspects of both theoretical physics and deep learning. Knowing symmetries of a physical problem is often the first step towards its solution. In machine learning, models that exploit symmetries of their data domain are most successful at their tasks, as exemplified by the success of convolutional neural networks for a variety of problems (Goodfellow et al., 2016). Recently, many deep learning papers have explored the concept of symmetry using tools from theoretical physics, e.g. (Mallat, 2016; Bronstein et al., 2017; Cohen et al., 2018; Higgins et al., 2018).

In most cases, these works assumed that the symmetries of the problem were manifest and built neural architecture that respected those symmetries, in order to learn more efficiently. In some situations, however, the symmetries of the problem are hidden from us, and much of the work is done to uncover those symmetries. A famous example is Kepler's inference from astronomical data that planetary orbits form ellipses with the sun at one focus. The fact that orbits in an inverse square law of force generically close

is a consequence of a subtle symmetry of the problem that gives rise to the conserved Laplace–Runge–Lenz vector (Goldstein et al., 2013).

Here, we present a data–driven way to learn such symmetries. While we concentrate here on models of Hamiltonian mechanics, we hope that the tools we develop will inspire research into symmetry learning more generally.

Learning a symmetry means learning a transformation from the original physical variables to a new set of variables in which the symmetry is manifest. Neural models describing bijective mappings are the subject of recent work on normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2014; 2016) and RevNets (Gomez et al., 2017; Jacobsen et al., 2018). Normalizing flows are usually constructed to have tractable Jacobians. In Hamiltonian mechanics *symplectic* (or *canonical*) transformations have a special role (Saletan & José, 1998). Such transformations are volume preserving but have further restrictions (see equation (2) below), and so require new network architectures. Hamiltonian time evolution is itself a symplectic transformation, so these methods may be applied to neural variational inference with Hamiltonian Monte Carlo (HMC) (Neal, 2012), discussed in several recent papers (Salimans et al., 2014; Wolf et al., 2016; Levy et al., 2017; Caterini et al., 2018; Hoffman et al., 2019).

In this work, we will focus on integrable models, which have nontrivial symmetries, as a test case for symmetry discovery. The organization of the remainder of this paper is as follows. In the next section, we introduce some concepts from classical integrable systems that we will use. Section 3 introduces our new architecture and learning algorithm, while section 4 contains experiments on three integrable models. Finally, section 5 provides a discussion of the results. Supplementary details are contained in the appendices.

2. Classical integrable systems

2.1. Hamiltonian dynamics and canonical transformations

Classical mechanics is the realm of Hamiltonian dynamics (Saletan & José, 1998). In the simplest case that we address here, motion occurs in a phase space $\mathbb{R}^{2n}$ of positions $q \in \mathbb{R}^n$ and momenta $p \in \mathbb{R}^n$. The dynamics is governed by Hamilton's equations, which are derived from a Hamiltonian

---

[1]Qualcomm AI Research, Amsterdam, NL. [2]TCM Group, Cavendish Laboratory, University of Cambridge, J. J. Thomson Ave., Cambridge CB3 0HE, UK. Correspondence to: Roberto Bondesan <r.bondesan@gmail.com>.

function $H : \mathbb{R}^{2n} \to \mathbb{R}$. For $x = (q, p)$ these read:

$$\dot{x} = \Omega \nabla_x H , \quad \Omega = \begin{pmatrix} 0 & \mathbf{1}_n \\ -\mathbf{1}_n & 0 \end{pmatrix} . \qquad (1)$$

For example, the harmonic oscillator with unit frequency is described by $H = (p^2 + q^2)/2$, and the equations of motion $\dot{q} = p, \dot{p} = -q$ describe circular orbits in the phase plane.

The skew-symmetric matrix Ω is called a symplectic form, and allows one to define symplectic (or canonical) transformations of phase space, as those maps f whose Jacobian matrix J_f is an element of the linear symplectic group $\text{Sp}_{2n}(\mathbb{R})$ at each point of its domain:

$$J_f^T \Omega J_f = \Omega . \qquad (2)$$

Since $\text{Det}(J_f) = +1$, volume is conserved. Equation (2) is however much more restrictive: the sum of (signed) areas in each $q_j - p_j$ plane is preserved (figure 1).

Given u, v scalar valued functions on phase space, their Poisson bracket is defined as $\{u, v\} \equiv (\nabla_x u)^\top \Omega \nabla_x v$. and is a symplectic invariant. With this notation, Hamilton's equations are $\dot{x} = \{x, H\}$, and time evolution is itself a symplectic transformation (Arnol'd, 2013).
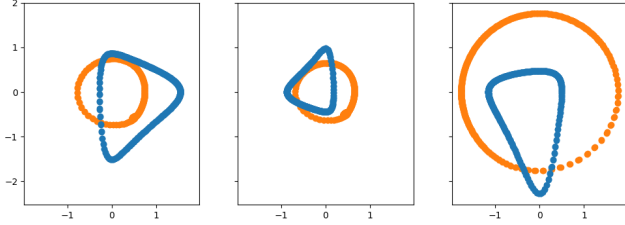

Figure 1. Projections of the original (blue) and transformed (orange) trajectories of the Kepler Hamiltonian (equation (17) with $k = -1$) onto the three $q_j - p_j$ phase planes. Note that the sum of enclosed areas is the same for the original and transformed trajectories.

2.2. Integrable models

A conserved quantity of a dynamical system is constant in time, and thus Poisson–commutes with the Hamiltonian, and constitutes a symmetry of the problem. For example, in the celebrated Kepler problem describing the motion of two planets attracted by a gravitational force which depends only on the distance between the planets, H commutes with the angular momentum, which generates rotations of the relative coordinate, and is a conserved quantity of the dynamics (Goldstein et al., 2013).

Integrable systems are those which have a number of mutually commuting and independent integrals of the motion that equals n, half the phase space dimension. The Liouville–Arnold theorem states that (compact) motion is confined

to torii parametrized by angles $\varphi_1, \ldots, \varphi_n$ and there exists a symplectic transformation $\mathcal{T}^{-1}$ from the original coordinates q, p to new coordinates φ, I, where I are called actions and are the conserved quantities of the problems (Arnol'd, 2013). In the action angle coordinates, equation (1) therefore reads:

$$\dot{\varphi} = \partial_I K = \text{const.} , \quad \dot{I} = -\partial_\varphi K = 0 , \qquad (3)$$

where the transformed Hamiltonian

$$K = H \circ \mathcal{T} , \qquad (4)$$

is independent of the angles. Finding explicit action-angle transformations is a challenging task and while a lot of progress has been made constructing integrable systems from algebraic or geometric principles (Babelon et al., 2003), there is no general algorithm to construct higher integrals of the motion given an integrable Hamiltonian. Learning such a transformation is the goal of this work.

We will work with the Cartesian coordinates $(\hat{q}_i = \sqrt{2I_i} \cos(\varphi_i), \hat{p}_i = \sqrt{2I_i} \sin(\varphi_i))$ and denote by T the symplectic map

$$T : (\hat{q}, \hat{p}) \mapsto (q, p) . \qquad (5)$$

For example, action-angle variables for the harmonic oscillator are the symplectic polar coordinates $(\arctan(p/q), (p^2 + q^2)/2)$, so that T is the identity, and the only conserved quantity is the energy. In general T will be such that complex trajectories in the (q, p) phase space get mapped to circles in the $(\hat{q}_i, \hat{p}_i)$ planes where φ_i is the angular coordinate and I_i is half the squared radius of the circle.

In this work we will learn neural parametrizations of T for three paradigmatic integrable models: (1) the Kepler model of two planets interacting with gravitational force (Goldstein et al., 2013); (2) the Neumann model of n oscillators with positions in $\mathbb{R}^n$ constrained to the $(n-1)$–dimensional sphere (Babelon et al., 2003); (3) the Calogero-Moser (CM) model of a chain of n particles with inverse square interaction potential (Moser, 1976). The Hamiltonians and their conserved quantities are described in appendix B.

3. Deep symplectic flows

We have reformulated the task of learning symmetries in integrable models as that of learning the map T which transforms a circular trajectory $(\hat{q}_i(t), \hat{p}_i(t))$ to the complex trajectory of the original model $(q_i(t), p_i(t))$. We now describe how to parametrize and learn such a transformation.

3.1. Parametrization

Here we will adapt recent results on normalizing flows to provide symplectic versions of popular invertible layers such

as additive coupling (Dinh et al., 2014), batch normalization (Dinh et al., 2016) and invertible linear transformations (Kingma & Dhariwal, 2018). Our parametrization of T is given by stacking m blocks of these three layers. We now describe each layer.

3.1.1. SYMPLECTIC ADDITIVE COUPLING

The additive coupling layer introduced in (Dinh et al., 2014) partitions the inputs $z = (z_A, z_B)$ and outputs $x = (x_A, x_B)$ with $x_A = z_A, x_B = z_B + \text{NN}(x_A)$, where the shift function NN is an arbitrary neural network. If we now identify A, B subsystems as q, p respectively, we have the following layer $L : (q, p) \mapsto (Q, P)$:

$$(Q, P) = (q, p + \text{NN}(q)), \quad (q, p) = (Q, P - \text{NN}(Q)). \tag{6}$$

Symplecticity of the transformation imposes further irrotationality:

$$\partial_i \text{NN}_j = \partial_j \text{NN}_i. \tag{7}$$

This constraint may be handled by setting $\text{NN}(q) = \nabla F(q)$, where $F : \mathbf{R}^n \to \mathbf{R}$ is parametrized by a neural network. This gives the traditional leapfrog update used in HMC.

While conceptually simple, this approach is computationally expensive, requiring $O(n^2)$ time in the backward pass of the network. A cheaper approach is to use a multilayer perceptron with three layers and constrained weight matrices $W^a \, a = 1, 2, 3$. In appendix A we show that equation (7) is satisfied if

$$W^3 = W^{1\top}, \quad W^2 = \text{diag}(w_1^2, \dots, w_{n_2}^2). \tag{8}$$

We call this architecture irrotational MLP. The analysis can be generalized, but we took this simple architecture for most of our experiments. Geometrically, W^1 embeds q into a higher dimensional space and $W^{1\top}$ maps the embedding back to the original space after it has been scaled by W^2, whose sign controls whether the map is orientation preserving.

3.1.2. SYMPLECTIC LINEAR

The additive coupling leaves q unchanged, and we introduce the symplectic linear layer to mix p, q so that deeper additive couplings act on all phase space coordinates. To parametrize a symplectic matrix $S \in \text{Sp}_{2n}(\mathbb{R})$, we use the pre–Iwasawa decomposition (de Gosson, 2006):

$$S = NAK = \begin{pmatrix} \mathbf{1} & 0 \\ M & \mathbf{1} \end{pmatrix} \begin{pmatrix} L^\top & 0 \\ 0 & L^{-1} \end{pmatrix} \begin{pmatrix} X & -Y \\ Y & X \end{pmatrix}, \tag{9}$$

with

$$M = M^\top, \quad X^\top Y = Y^\top X, \quad X^\top X + Y^\top Y = \mathbf{1}. \tag{10}$$

To parametrize K, we note that it is the realification of the unitary $X + iY$ and can be written as a product of Householder reflections, parametrized by a vector v:

$$R_v = \mathbf{1} - 2\frac{vv^\dagger}{||v||^2} \in U_n, \tag{11}$$

and a diagonal matrix of phases

$$U = \text{diag}(e^{i\phi_i}). \tag{12}$$

We refer to (Cabrera et al., 2010; Tomczak & Welling, 2016) for background on Householder reflections.

Note that the complexity of applying both R_v and U to a vector (q, p) is $O(n)$. To keep the complexity of the whole layer $O(n)$ we take $r = O(1)$ Householder reflections and further take $L = \text{diag}(L_i)$ and $M = \text{diag}(M_i)$.

3.1.3. ZERO CENTER

The zero center layer is defined by the transformation:

$$\begin{cases} Q = q - \mu^q + \alpha \\ P = p - \mu^p + \beta \end{cases}, \tag{13}$$

where μ is the mean given during training as the batch mean, and during testing by a weighted moving average accumulated during training, as in the batch normalization layer – see (Dinh et al., 2016) for its usage in normalizing flows. α, β are learnable offsets. The zero center layer normalizes its input and allows one to study deeper architectures, and is a restricted version of batch normalization compatible with symplectic invariance. (The full version of batch normalization indeed scales by the variance of each feature and does not preserve areas.)

3.2. Learning algorithm

According to the discussion in section 2, the map T^{-1} is determined by requiring that the original trajectory $(q_i(t), p_i(t))$ is mapped to circles $(\hat{q}_i(t), \hat{p}_i(t))$. If the trajectory is sampled at τ time steps t_k, such T minimizes the following loss, which encourages the distance from the origin of neighbouring points to be the same:

$$\ell = \frac{1}{n\tau} \sum_{k=1}^{\tau} ||r_k - r_{k+1}||^2, \quad r_k = \hat{q}(t_k)^2 + \hat{p}(t_k)^2. \tag{14}$$

A non-invertible or non-volume preserving network could minimize equation (14) trivially by collapsing the trajectories to zero or very small volumes in the transformed phase space: the symplecticity of the transformation is essential.

We therefore consider a learning algorithm that takes as input a batch of trajectories and minimizes the loss above
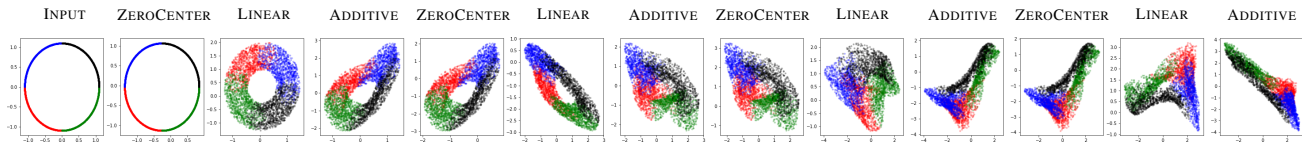
Figure 2. Visualization of the transformations done by each layer of T along a given $q_j - p_j$ phase plane for the Neumann model. The input points (left) belong to a cycle of the Liouville–Arnold torus. Colors refer to which quadrant of the input plane a point comes from.

averaged over the batch. Practically, we compute the trajectories by solving the original equations of motions using a Runge-Kutta solver, and we perform stochastic gradient descent parameter updates using the Adam optimizer. We shuffle randomly the trajectories at every epoch, which ensures that we compare distant points in equation (14), so that all deviations from a circular shape are penalized in the same way.
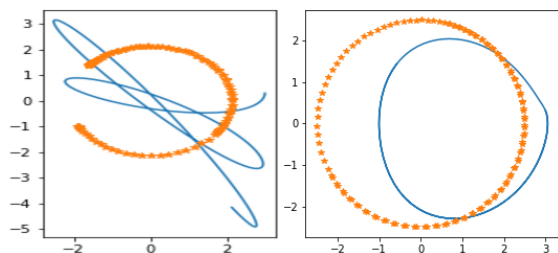
4. Experiments

Figure 3. Pull back of trajectories (thin blue line) under the map T^{-1} learned. Each model has phase space of dimension $2n = 6$ and here a single $q - p$ phase plane is selected for illustration.

We now present the results of experiments on the three models defined in appendix B. The code used is available at **https://github.com/rbondesan/CanonicalFlows/**, and we refer to appendix C for details of network architecture and other training hyperparameters.

We first discuss the representation capacity of deep symplectic flows. Figure 3 shows the pull-back of the trajectories under a network T^{-1} composed of $m = 4$ blocks defined in section 3.1. The parameters in T^{-1} are learned by running the algorithm of section 3.2 to convergence and feeding it with a single trajectory sampled at $\tau = 128$ time steps. This shows that our model and algorithm can learn the action–angle map for both periodic (Kepler and Calogero-Moser) and quasi-periodic (Neumann) models.

We next investigate the generalization of our learning algorithm. By this we mean how well the learned symplectic flow can map unseen trajectories to circles. We present here results for the Neumann model with $n = 3$ oscillators. We consider a batch of trajectories, one for each

Table 1. Test loss $\ell' = \ell \times 10^5$ for trajectories in the Neumann model at radius r. Bold text denotes radii of training trajectories.

r	2	**3**	4	**5**	6	**7**	8
ℓ'	6.5	**3.7**	23.2	**12.4**	117.5	**23.4**	141.6

radius $r = 2, 3, \ldots, 8$ of the sphere on which the positions of the oscillators move. Table 1 reports the loss of equation (14) evaluated over these trajectories for a symplectic flow T learned by considering only the subset $r = 3, 5, 7$ as training data. While the points in the training set have the smallest values compared to neighbouring radii, the fact that the loss is of the same order across this range of trajectories, indicates that the model generalizes beyond the training points. To further substantiate this claim, we show in figure 4 the pull-back of the trajectories under T^{-1} for both the trajectories seen and unseen by the training algorithm.

The map T thus learned can be used as a generative process for the physical system trajectories, as illustrated in figure 2. Interestingly, this allows one to visualize the points in phase space that correspond to a given Liouville–Arnold torus. Further, by varying the circles radii one can interpret the effect of the learned symmetries on the system under consideration.

5. Discussion

The learning algorithm discussed so far relies on being able to solve the equations of motion. This was done here by numerical integration, and adds questions of convergence and stability of the ODE solver on top of those of the learning algorithm. We remark that there two possible ways to improve this. The first is to exploit integrability of the models to solve the motion analytically, which is however not a trivial matter and typically requires uncovering a Lax pair formulation of the problem (Babelon et al., 2003). The second is to use a different learning algorithm. For example, one could minimize a loss that encourages the transformed Hamiltonian to be angle-independent (recall equation (3)), or one could minimize the Kullback–Leibler (KL) divergence between the canonical density associated to the transformed Hamiltonian and that of a base distribution. Both alternatives are analyzed in appendix D.
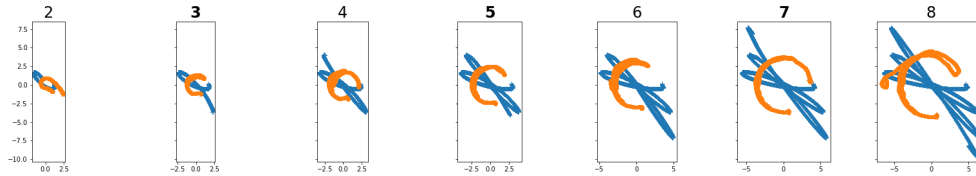
Figure 4. Pull back of trajectories (thin blue line) for the Neumann model at radius r, indicated by the figures titles. Bold text denotes radii of training trajectories. A single $q - p$ phase plane is selected for illustration.

While we have concentrated here on integrable models, we expect our methods to be applicable to find almost conserved quantities in models close to integrable ones, such as the celebrated Fermi–Pasta–Ulam–Tsingou chains (Gallavotti, 2007). We thus expect the deep learning approach to classical mechanics presented here to be of practical relevance for solving physical problems, for example by finding integration schemes with smaller discretization errors.

References

Arnol'd, V. I. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

Babelon, O., Bernard, D., Talon, M. *Introduction to Classical Integrable Systems*. Cambridge Monographs on Mathem. Cambridge University Press, 2003. ISBN 9780521822671.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Cabrera, R., Strohecker, T., and Rabitz, H. The canonical coset decomposition of unitary matrices through Householder transformations. *Journal of Mathematical Physics*, 51:082101–082101, Aug 2010. doi: 10.1063/1.3466798.

Caterini, A. L., Doucet, A., and Sejdinovic, D. Hamiltonian variational auto-encoder. In *Advances in Neural Information Processing Systems*, pp. 8167–8177, 2018.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical CNNs. *arXiv preprint arXiv:1801.10130*, 2018.

de Gosson, M. *Symplectic Geometry and Quantum Mechanics*. Operator Theory: Advances and Applications. Birkhäuser Basel, 2006. ISBN 9783764375751.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation. *ArXiv e-prints*, October 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. *ArXiv e-prints*, May 2016.

Gallavotti, G. *The Fermi-Pasta-Ulam problem: a status report*, volume 728. Springer, 2007.

Goldstein, H., Poole, C., and Safko, J. *Classical Mechanics*. Pearson, 2013. ISBN 9781292026558.

Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pp. 2214–2224, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN 9780262035613.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. *arXiv preprint arXiv:1903.03704*, 2019.

Jacobsen, J.-H., Smeulders, A., and Oyallon, E. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

Jaynes, E. T. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Kingma, D. P. and Dhariwal, P. Glow: Generative Flow with Invertible 1x1 Convolutions. *ArXiv e-prints*, July 2018.

Levy, D., Hoffman, M. D., and Sohl-Dickstein, J. Generalizing hamiltonian monte carlo with neural networks. *arXiv preprint arXiv:1711.09268*, 2017.

Mallat, S. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150203, 2016.

Moser, J. Three integrable hamiltonian systems connected with isospectral deformations. In *Surveys in Applied Mathematics*, pp. 235–258. Elsevier, 1976.

Neal, R. M. MCMC using Hamiltonian dynamics. *ArXiv e-prints*, June 2012.

Perelomov, A. M. *Integrable systems of classical mechanics and Lie algebras*. Birkhäuser, 1990.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pp. 1530–1538. JMLR. org, 2015.

Saletan, J. and José, J. Classical dynamics: A contemporary approach, 1998.

Salimans, T., Kingma, D. P., and Welling, M. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. *ArXiv e-prints*, October 2014.

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

Wolf, C., Karl, M., and van der Smagt, P. Variational Inference with Hamiltonian Monte Carlo. *arXiv preprint arXiv:1609.08203*, 2016.

A. Symplecticity of irrotational MLP

We demonstrate that the stated form of the weights in equation (8) for a three layer MLP satisfies the irrotationality condition of equation (7). The Jacobian of the MLP is

$$\frac{\partial \text{NN}(x)}{\partial x} = W^3 D^2 W^2 D^1 W^1 \,, \qquad (15)$$

with $D^a = \text{diag}(\lambda^a)$, $\lambda^a = (\sigma^a)'(W^a x^a + b^a)$, where $W^a, b^a, \sigma^a, x^{a+1}$ are resp. $n_a \times n_{a-1}$ weights, bias, activation functions and activations of the layer a. The constraint of equation (7) then takes the form

$$W^3 D^2 W^2 D^1 W^1 = W^{1\top} D^1 W^{2\top} D^2 W^{3\top} \,. \qquad (16)$$

We further take $n_2 = n_3$, so that $D^2 W^2 D^1$ is square. The symmetry condition equation (16) means that $W^{1\top} = W^3$ and $D^2 W^2 D^1$ is symmetric. For generic λ^1, λ^2, the only solution is $W_{ij}^2 = 0$ for $i \neq j$. This verifies equation (8).

B. Hamiltonians and conserved quantities

We provide details of the three integrable models studied.

B.1. Kepler

The Hamiltonian of the Kepler model is

$$H_{\text{K}} = \tfrac{1}{2} \sum p_i^2 + \frac{k}{r}, \quad r = \sqrt{\sum q_i^2}. \qquad (17)$$

As well as the Hamiltonian H_{K} and the three components of the angular momentum $\mathbf{L} = \mathbf{q} \times \mathbf{p}$, the Kepler model has an additional conserved quantity called the *Laplace–Runge–Lenz vector* (Goldstein et al., 2013)

$$\mathbf{A} = \mathbf{p} \times \mathbf{L} + k\frac{\mathbf{q}}{r}. \qquad (18)$$

This gives a total of seven conserved quantities. However, a one dimensional trajectory in a six dimensional phase space can have at most five conserved quantities that specify the trajectory. Thus there must be two relations between these quantities. They are

$$\mathbf{A} \cdot \mathbf{L} = 0$$
$$\mathbf{A}^2 = k^2 + 2H_{\text{K}}\mathbf{L}^2 \qquad (19)$$

The existence of five independent conserved quantities, two more than the three required for integrability, mean that the trajectories form closed curves rather than filling the Liouville–Arnold torii. This situation is called *superintegrability*.

B.2. Neumann

The Neumann Hamiltonian is

$$H_{\mathrm{N}} = \tfrac{1}{4} \sum_{i,j} J_{ij}^2 + \tfrac{1}{2} \sum a_i q_i^2 \,, \quad J_{ij} = q_i p_j - q_j p_i. \quad (20)$$

An independent set of constants of motion are (Babelon et al., 2003)

$$I_i = q_i^2 + \sum_{i \neq j} \frac{J_{ij}^2}{a_i - a_j} \quad (21)$$

for k_α all different. The property $\{I_i, I_j\} = 0$ for $i \neq j$ is easily checked using $\{q_i, p_j\} = \delta_{ij}$. The Hamiltonian can be expressed as the linear combination

$$H_{\mathrm{N}} = \frac{1}{2} \sum_i a_i I_i. \quad (22)$$

Additionally,

$$\sum_i q_i^2 = \sum_i I_i, \quad (23)$$

showing that motion is confined to the sphere S^{n-1}.

B.3. Calogero–Moser

The Calogero–Moser model is given by

$$H_{\mathrm{CM}} = \tfrac{1}{2} \sum (p_i^2 + \omega^2 q_i^2) + \sum_{j < k} \frac{g^2}{(q_j - q_k)^2} \,. \quad (24)$$

The discussion of integrals of motion is facilitated by presenting the equations of motion in matrix form (Perelomov, 1990). Introducing the matrices

$$L^{\pm} = L \pm i\omega Q$$
$$Q_{ij} = q_i \delta_{ij}$$
$$L_{ij} = p_i \delta_{ij} + (1 - \delta_{ij}) \frac{ig}{q_i - q_j}$$
$$M_{ij} = g \left[\delta_{ij} \sum_{k \neq i} \frac{1}{(q_i - q_k)^2} - (1 - \delta_{ij}) \frac{1}{(q_i - q_j)^2} \right],$$
$$(25)$$

the equations of motion are equivalent to

$$\dot{Q} + i[M, Q] = L$$
$$\dot{L} + i[M, L] = -\omega^2 Q. \quad (26)$$

From this point it can be shown that the quantities

$$I_k = \mathrm{tr}\left[(L_+ L_-)^k \right] \quad (27)$$

are all conserved.

C. Details of experiments

We present here details of the experiments of section 4.

In figures 2 and 3 we used frequencies $a_1 = .1, a_2 = .2, a_3 = .3$ for the Neumann model and couplings $\omega^2 = g^2 = 1$ for the Calogero-Moser model. The network architecture used is composed of $m = 4$ blocks of zero-center, linear symplectic and symplectic additive coupling layers of section 3.1, applied in this order. The linear symplectic layer has $r = 2$ Householder reflections whose reflection vectors are initialized to be identical, so that their product is the identity. The irrotational MLP used in the additive coupling layer (see appendix A) has hidden dimension 512 and $\mathtt{tanh}$ activation functions.

The data reported in table 1 and figure 4 was produced by sampling at $\tau = 4096$ points solutions of the equations of motions between times 0 and 10. Training was done with the Adam optimizer using a mini-batch size of 128. We trained for 7395 epochs using a piece-wise learn rate schedule with iteration step boundaries $[20000, 50000, 100000]$ and values $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$.

D. Alternative algorithms

D.1. `dKdPhi`

The algorithm of section 3.2 used numerical solutions of the equations of motion for a batch of initial conditions. This can be expensive for large n and also introduces numerical errors and questions of convergence of the ODE solver. We here present an alternative procedure. Recall that the transformed Hamiltonian K of equation (4) satisfies $\nabla_\varphi K = 0$. We therefore could find T by minimizing the expectation of the norm of $\nabla_\varphi K$. However, the natural expectation is with respect to the canonical density of the model, which is $\propto \exp(-K(I))$ and itself unknown. We propose to replace the canonical density with the exponential density

$$\rho(\varphi, I) = (2\pi)^{-n} \exp - \sum_i I_i \,, \quad (28)$$

where the numerical prefactor comes from the uniform density of the angles. This choice is motivated by the fact that integrable models are expected to be described by the generalized Gibbs ensemble when n is large (Jaynes, 1957), and which corresponds to an exponential distribution for the actions. Therefore, we propose the following alternative loss:

$$\ell = \mathbb{E}_{(\varphi, I) \sim \rho} \{ ||\nabla_\varphi K||^2 \} \,, \quad (29)$$

where we estimate the expectation over minibatches.

For illustration, we present some results using this algorithm for the Neumann model. For simplicity, we trained the model by sampling uniformly in the angles but fixing a

single value of the actions, effectively replacing the exponential measure in ρ with a Dirac measure, which physically amounts to restricting sampling to a single Liouville–Arnold torus. The model parameters and network architecture are similar to those of appendix C. To test the algorithm, we solve the Neumann equations of motion for an initial condition corresponding to the value of the actions used for training, and check that the inverse map is able to trivialize the trajectory by mapping it to a circle. We stress that differently from section 4, here the trajectories are not inputs to the training algorithm, which is fully unsupervised. The result is presented in figure 5, and validates the procedure.
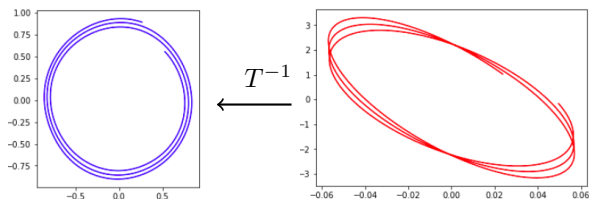

Figure 5. Pull back of Neumann trajectories (right) under the map T^{-1} learned by minimizing equation (29) for a single $q - p$ phase plane.

While not relying on numerical solutions of the equations of motion is appealing, we noticed that this algorithm takes a longer time to converge than that of section 3.1. This is partly due to the presence of a derivative in the loss, which increases the computational cost.

D.2. Normalizing flows and KL minimization

The algorithm of appendix D.1 relies on the exponential ansatz for the sampling measure, which lacking rigorous results about its validity, can be not optimal in practice. Also, while one learns directly the transformed Hamiltonian K, one does not know how to sample efficiently from it. Both issues can be resolved if we take a learnable density

$$\rho(I, \varphi) = \rho_0(F^{-1}(I), \varphi) \left| \text{Det}(\partial_I F^{-1}(I)) \right|, \qquad (30)$$

where F is a normalizing flow (Dinh et al., 2016). The base distribution ρ_0 can be taken to be the exponential distribution.

In this formulation, a natural loss to consider is the following KL divergence:

$$\ell = \mathbb{E}_{(u,\varphi) \sim \rho_0} \left(\text{KL}(\rho_0, \pi \circ T \circ F) \right) \qquad (31)$$

$$\sim \frac{1}{N} \sum_{i=1}^{N} H(T(F(u_i), \varphi_i)) + \text{const}, \qquad (32)$$

where $\pi \propto \exp(-H)$ is the canonical density associated to the Hamiltonian H. We leave the numerical study of the minimization of this loss for future work.