

Inference of Hamiltonian and Lie-Poisson structures from vector field samples

Jason E. Frank

Georg A. Gottwald

October 3, 2026

Abstract

We present a method for the data-driven inference of Hamiltonian and Lie-Poisson dynamical systems from vector-field samples, in which both the Hamiltonian and the underlying Poisson tensor are estimated jointly, rather than assuming the structure is known in advance. The Hamiltonian is represented as a linear combination of random feature maps with fixed, randomly drawn internal weights, which renders its estimation a linear regression problem for any given Poisson structure. We treat three cases of increasing generality: a known Poisson tensor, an unknown constant invertible (symplectic) structure matrix, and the case of Lie-Poisson and affine Poisson structures, in which the structure matrix is linear in the state. For the constant case, the Hamiltonian and structure matrix are recovered jointly as the solution of an eigenvalue problem that avoids the trivial zero solution induced by the scaling symmetry of Hamiltonian systems. For Lie-Poisson and affine Poisson systems we use an alternating scheme: an initial Hamiltonian estimate is obtained from a generalized eigenvalue problem for an approximate first integral, and the structure constants are then fit by constrained nonlinear least squares, enforcing the Jacobi identity and, for affine structures, the associated cocycle condition, via continuation in a penalty parameter. Orthogonal matching pursuit selects a compact, well-conditioned subset of random features. Because the learned Hamiltonian decomposes into single-feature terms, the resulting model admits an explicit Poisson/symplectic splitting integrator, which we use to validate the learned systems over long-time integrations via Poincaré sections. We demonstrate the approach on the five-component Lorenz-86 model and a Kirchhoff-type rigid-body system, recovering Hamiltonians, structure constants, and Casimir invariants to high accuracy, and showing convergence of the inferred Poincaré sections to the correct topology of invariant tori as the number of random features increases.

1 Introduction

Recent years have seen a surge in interest in developing methods to perform data-driven modelling of dynamical systems [23, 31, 34, 36]. These attempts can broadly be split into two categories, each coming with their own set of applications and aims. One class of methods is concerned with learning surrogate models of a dynamical system that propagate an initial condition to the state at later time. The quest for such surrogate models

is driven by computational factors: executing a trained surrogate model offers significant computational advantages over numerically integrating the full high-dimensional often stiff complex dynamical system. A prominent example is weather forecasting where significant advances have recently been achieved by exploiting machine learning [3, 21, 35]. Recent years have seen advancement in finding surrogate maps for Hamiltonian systems using neural networks, including Lie-Poisson neural networks, [10, 37] who learn a Poisson map to advance the system, given known Poisson bracket structure, and recent work by Hansen et al. [16], who learn a discrete Lagrangian model which they extend to a discrete Lagrangian construction on Lie groups in [9]. The second strand of data-driven modelling is concerned with system identification where, given data, one aims to estimate the vector field of the dynamical system. We are concerned here with the latter class of system identification. Whereas system identification has been successfully achieved for large classes of dissipative dynamics, remarkably even with computationally relatively simple regression methods such as SINDy [5], the class of Hamiltonian dynamical systems have proved to be resistant to an efficient estimation. Methods that enjoy success for dissipative systems, including reservoir computers, are unable to preserve the geometric structure innate in Hamiltonian dynamics [33, 43]. The failure to encode the innate geometrical structure of a Hamiltonian system prevents the trained vector fields being used for long term integration, and typically implies nonphysical energy drain. A promising strategy to overcome these limitations is to approximate the Hamiltonian function rather than the full vector field [13, 15, 7, 2, 19, 18]. This, however, requires the knowledge of the symplectic structure matrix and excludes the important case of Poisson systems with a state-dependent structure matrix. For the particular classes of metriplectic systems or systems with the GENERIC structure neural networks were shown to be successful [22, 43]. Such systems are driven by two integral quantities, a Hamiltonian representing total energy and a single Casimir that is conserved in the dissipation-free limit but grows monotonically in general. In [22] fixed structure for the symmetric and antisymmetric tensors is assumed, whereas in [43] the bracket structures may be learned. However, neither article enforces the Jacobi identity, so the skew-symmetric brackets are generally not Poisson. Using Gaussian process kernel regression [18] approximate the Hamiltonians for Poisson systems, given known Poisson structure.

How to estimate simultaneously the Hamiltonian and the Poisson structure remains an open problem. Here we show that this can be achieved for a particular class of Poisson systems, namely, Lie-Poisson and affine Poisson systems for which the structure matrix is linear in the state variables. Hamiltonian system with Lie-Poisson structure (6) admit a representation as a dynamical system on the dual to a Lie algebra, and consequently have a rich mathematical theory. They cover a large range of Hamiltonian systems including the free rigid body, ideal incompressible fluids, magnetohydrodynamics, and Vlasov-type kinetic equations [29, 17].

A particularly attractive data-driven modelling framework are random features [38]. Random feature maps [38] have recently been shown to perform very well for learning surrogate maps for dynamical systems [12, 30, 25, 27, 28]. They are particularly at-

tractive as they render the estimation problem as a regression problem, avoiding costly optimization procedures typical for deep neural networks. Such simple network architectures often outperform more complex machine learning approaches achieving better forecast skill with less hyperparameter tuning and orders of magnitude smaller networks [27, 42]. Here we use random features to perform system identification. We show that random feature models allow for an effective bilevel regression to estimate both the Hamiltonian and the structure matrix.

We represent the Hamiltonian as a linear combination of random feature maps. We solve the joint regression for the Hamiltonian and structure constants for symplectic, Lie-Poisson and affine Poisson structures given vector field sample data. Ensuring the structure constants adhere to the Jacobi identity places strong constraints on the minimization problem, which we address using a continuation strategy within a nonlinear least squares solver. To test the quality of the learned systems we compute Poincare sections, requiring long numerical integrations. Our main conclusions are (1) that representation of the Hamiltonian as a random feature map, in combination with hit-and-run feature selection and optimal matchin pursuit admits a computationally efficient least-squares regression, and (2) combining this approach with constrained regression for the structure constants allows us to infer all of the necessary Hamiltonian structure for accurate system identification and long-term prediction.

The paper is organized as follows. Section 2 recalls the dynamical setting of Hamiltonian systems and in particular of Lie-Poisson systems. In Section 3 we provide a brief overview of random feature maps and develop the random feature map representation for Hamiltonian systems. In particular, we cover the case when the Poisson structure is known, the case when the Hamiltonian is learned together with a constant symplectic structure, and finally the case of learning both the Hamiltonian and structure constants for the cases of Lie-Poisson and affine Poisson structures. In Section 4 we show that random feature map Hamiltonians admit Poisson splitting methods for their numerical integration. Applying these geometric integrators, we show in Section 5 that our trained random feature models preserve the underlying geometrical structure and are capable of accurately reproducing the dynamics of the four and five-dimensional Lorenz-86 model and a six-dimensional Kirchhoff rigid body model. We conclude in Section 6 with a discussion and an outlook.

2 Hamiltonian dynamical systems and Lie-Poisson systems

System identification is concerned with estimating the vector field $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ of a dynamical system

$$\dot{x} = f(x), \quad x(t) \in \mathbb{R}^d. \quad (1)$$

We are concerned here with Hamiltonian systems [29, 17]. A Hamiltonian system of differential equations on $\mathbb{R}^d$ is defined by a differentiable function $H : \mathbb{R}^d \rightarrow \mathbb{R}$ and a symplectic operator J , i.e. a nondegenerate, skew-symmetric operator $J \in \mathbb{R}^{d \times d} : J^\top =$

$-J$:

$$\dot{x} = J\nabla H(x). \quad (2)$$

The Poincaré theorem establishes that Hamiltonian systems are endowed with the geometric property of symplecticity.

Symplectic structure places a strong constraint on the dynamics, implying volume preservation, conservation of the Hamiltonian, as well as robustness of integrable structures under small perturbations (KAM theory). Furthermore, excellent symplectic numerical methods are available for simulating Hamiltonian systems, which provide long-time preservation of the above properties [14, 24, 41]. Any data-driven method estimating the vector field of a Hamiltonian dynamical system hence should preserve the underlying geometric structure and needs to explicitly infer the Hamiltonian structure from sample data.

Nondegeneracy and skew-symmetry of J together imply that d must be even. In the canonical case that arises naturally in Newtonian mechanics, the structure matrix has the specific form

$$J = \begin{bmatrix} 0 & I_{d/2} \\ -I_{d/2} & 0 \end{bmatrix}, \quad (3)$$

where $I_{d/2}$ is the identity matrix of dimension $d/2$. The phase space coordinate x is partitioned $x = (q, p)$, where $q, p \in \mathbb{R}^{d/2}$ are the conjugate coordinates and momenta, respectively. On the other hand, if $L \in \mathbb{R}^{d \times d}$ is an invertible matrix, we may define $\tilde{x} = Lx$, $\tilde{H}(\tilde{x}) = H(L^{-1}\tilde{x})$ and $\tilde{J} = LJL^\top$, and the system (2) transforms to

$$\dot{\tilde{x}} = \tilde{J}\nabla\tilde{H}(\tilde{x}),$$

where in general $\tilde{J}$ is a full, skew-symmetric matrix, that typically is not canonical.

More general Hamiltonian systems may have a Poisson structure, defined by a Poisson bracket, for which the matrix representation yields a system of the form

$$\dot{x} = B(x)\nabla H(x), \quad (4)$$

where d may be even or odd, and $B(x)$ is a Poisson tensor: a local matrix function of x that must satisfy both the skew-symmetry requirement $B^\top(x) = -B(x)$ and the Jacobi identity:

$$\sum_{\ell} \frac{\partial B_{ij}}{\partial x_{\ell}} B_{\ell k} + \frac{\partial B_{jk}}{\partial x_{\ell}} B_{\ell i} + \frac{\partial B_{ki}}{\partial x_{\ell}} B_{\ell j} = 0, \quad \forall x \in \mathbb{R}^d, \forall i, j, k \in \{1, \dots, d\}. \quad (5)$$

When $B(x)$ is degenerate (if d is odd, for instance), then there may be one or more Casimir functions $Z_i(x)$ satisfying $B(x)\nabla Z_i(x) = 0$. Such functions are first integrals of (4) for any Hamiltonian $H(x)$.

The Poisson tensor $B(x)$ is characterized by the Darboux theorem, which states that if $B(x)$ has rank $2r \leq d$ at x , there exists a local change of variables $\tilde{x} = S(x)$ under which

$B(x)$ takes the degenerate canonical form

$$J = \begin{bmatrix} 0 & I_r & 0 \\ -I_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

A Lie-Poisson structure is defined by a Poisson tensor that is linear in x ,

$$B(x)_{ij} = \sum_{k=1}^d C_{ij}^k x_k. \quad (6)$$

The skew-symmetry of $B(x)$ translates to the conditions $C_{ij}^k = -C_{ji}^k$ for all $i, j \in \mathbb{R}^d$. The Jacobi identity (5) leads to the conditions

$$\sum_{\ell=1}^d C_{ij}^\ell C_{\ell k}^m + C_{jk}^\ell C_{\ell i}^m + C_{ki}^\ell C_{\ell j}^m = 0, \quad i, j, k, m = 1, \dots, d. \quad (7)$$

Under a homogeneous linear transformation $\tilde{x} = S(x) = Lx$, with L an invertible matrix, the Lie-Poisson structure constants transform as

$$\tilde{C}_{ij}^k = L_{ii'} L_{jj'} L_{k'k}^{-1} C_{i'j'}^{k'},$$

which again satisfy (7).

An affine Poisson system is one of the form (4) with a Poisson tensor that is the sum of a Lie-Poisson tensor and a constant skew-symmetric tensor

$$B(x)_{ij} = \sum_k C_{ij}^k x_k + J_{ij}. \quad (8)$$

In this case, the structure constants C must be skew-symmetric and satisfy (7), and in addition J must be skew-symmetric and satisfy the cocycle condition

$$\sum_{\ell=1}^d C_{ij}^\ell J_{\ell k} + C_{jk}^\ell J_{\ell i} + C_{ki}^\ell J_{\ell j} = 0, \quad \forall i, j, k = 1, \dots, d. \quad (9)$$

Affine Poisson structure is again robust under affine linear transformations, i.e. under $\tilde{x} = S(x) = Lx + y$ for invertible L .

3 Random feature maps

Random feature maps can be viewed as a single layer feedforward network in which the internal weights are fixed. Rather than being learned as done in typical feedforward neural networks, in random feature maps the internal weights are drawn from some prescribed random distribution and held fixed. This allows for the estimation of the

outer weights by simple least square regression with an explicit analytical expression rather than having to learn them via computationally costly back propagation.

Consider a bounded subset $\mathcal{D} \subset \mathbb{R}^d$ upon which we wish to approximate an unknown function $f : x \mapsto \mathbb{R}^d$ for $x \in \mathcal{D}$; this includes our case of interest where $f(x)$ denotes a vector field $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$. We remark that it is easily generalized to functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Given internal weights $W_{\text{in}} \in \mathbb{R}^{K \times d}$ and biases $b_{\text{in}} \in \mathbb{R}^K$ of the internal layer drawn from some user-defined distribution, random features $\phi(x) \in \mathbb{R}^K$ are generated by an affine linear transformation of $x \in \mathbb{R}^d$ followed by a nonlinear activation function σ , that acts component-wise on vectors, as

$$\phi(x) = \sigma(W_{\text{in}}x + b_{\text{in}}). \quad (10)$$

We shall use, as often done, $\sigma(z) = \tanh(z)$. The random feature map $\hat{f}(x)$ approximating a general function $f(x)$ is provided by a linear combination of the K features as

$$\hat{f} : x \mapsto W \tanh(W_{\text{in}}x + b_{\text{in}}), \quad (11)$$

where the external layer weights $W \in \mathbb{R}^{d \times K}$ are to be learned to minimize the data mismatch $\|\hat{f}(x) - f(x)\|^2$ over the training data. In particular, we assume we are given M sample data in the sample space $\mathcal{D}$

$$X^{(\mu)} \in \mathcal{D}, F^{(\mu)} : \mathcal{D} \rightarrow \mathbb{R}^d, \quad \mu = 1, \dots, M,$$

where $F^{(\mu)} = f(X^{(\mu)})$. Introducing the design matrices $\Phi \in \mathbb{R}^{K \times M}$ and $F \in \mathbb{R}^{d \times M}$, where the μ th column consists of $\phi(X^{(\mu)})$ and $F^{(\mu)}$, respectively, the estimation of the external weights is given as the minimizer of the following regularized loss function,

$$\arg \min_W \|W\Phi(X) - F\|_F^2 + \beta \|W\|_F^2, \quad (12)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The regularization parameter β is a hyper-parameter which requires tuning. The solution to the ridge regression problem (12) is explicitly given by

$$W = F\Phi^\top \left(\Phi\Phi^\top + \beta I \right)^{-1}. \quad (13)$$

Random feature maps enjoy the universal approximation property, stating that in the limit $K \rightarrow \infty$ there exist internal weights W_{in} and b_{in} such that they approximate a continuous function arbitrarily well [6, 32, 1, 39]. The degree of approximation sensitively depends on the internal weights. If the internal weights are, for example, chosen for the $\tanh$ -activation function such that they evaluate to either ± 1 for all $x \in \mathcal{D}$, then the approximation capability is clearly significantly reduced compared to more expressive functions. Recently there has been interest in finding approximate methods to choose the internal parameters to increase the approximation quality of random feature maps

given a fixed feature dimension K [25, 8, 28]. We adopt here the strategy proposed in [28] and employ a hit-and-run algorithm to draw the internal parameters such that for given training data the weights do not project the data into either the saturated region of the tanh-function, i.e. evaluating to either ± 1 , or into the approximately linear region of the tanh-function. In the former case, the random features cannot differentiate between different data inputs whereas in the latter case they would reduce to a linear model which is incapable of encoding a nonlinear dynamical system.

In practice the number of random features can be reduced using orthogonal matching pursuit (OMP) as described in [40]. Batch-OMP recursively constructs a subset of features upon which the orthogonal projection of the residual is maximal. OMP terminates when the next best candidate feature is contained in the span of selected features to within a chosen relative tolerance (10^{-10}), signalling numerical linear dependence of remaining features.

In the case when X are samples of state space variables x and F are samples of the associated vector field of a dynamical system $\dot{x} = f(x)$, one may be tempted to directly apply (13) to obtain an approximation of the vector field $\hat{f}(x) = W \tanh(W_{\text{in}}x + b_{\text{in}})$ with $W \in \mathbb{R}^{d \times K}$. This strategy, however, would fail for an underlying Hamiltonian system where $f(x) = B(x)\nabla H(x)$ in the sense that the particular geometric structure will not be preserved. Rather than expressing the vector field by random features, we follow [19] and instead express the Hamiltonian $H(x)$ by random features as

$$\hat{H}(x) = \alpha^\top \tanh(W_{\text{in}}x + b_{\text{in}}) \quad (14)$$

with learnable weights $\alpha \in \mathbb{R}^K$ and internal weights $W_{\text{in}} \in \mathbb{R}^{K \times d}$ and biases $b_{\text{in}} \in \mathbb{R}^K$. Comparing with the notation of (11) we remark that $W = \alpha^\top$, i.e. a row vector for the case of a scalar valued function.

By expressing the Hamiltonian as a random feature map (14), the vector field can be expressed as a linear function of the external weights α_κ . Define

$$\psi_\kappa(x) = \text{sech}^2(w_\kappa^\top x + b_\kappa), \quad \kappa = 1, \dots, K, \quad (15)$$

where $w_\kappa^\top \in \mathbb{R}^d$ denotes the κ th row of $W_{\text{in}} \in \mathbb{R}^{K \times d}$ and b_κ is the κ th component of the biases $b_{\text{in}} \in \mathbb{R}^K$.

Introduce the associated design matrix $\Psi \in \mathbb{R}^{K \times M}$ with components $\psi_\kappa^{(\mu)} = \psi_\kappa(X^{(\mu)})$, $k = 1, \dots, K$, $\mu = 1, \dots, M$. Let $\psi^{(\mu)} = \psi(X^{(\mu)}) \in \mathbb{R}^K$ denote the μ th column of Ψ . The gradient of the Hamiltonian expressed as a random feature map with $\phi_\kappa(x)$ (10) with $\sigma = \tanh$, is then given by

$$\nabla \hat{H}(X^{(\mu)}) = \sum_{\kappa=1}^K \alpha_\kappa \psi_\kappa^{(\mu)} w_\kappa = W_{\text{in}}^\top \text{diag}(\alpha) \psi^{(\mu)}. \quad (16)$$

In the following we describe how to use random feature maps to learn three different classes of Hamiltonian dynamical systems of increasing complexity: (i) a generic Hamiltonian system with a given Poisson tensor $B(x)$, for which we only need to estimate

the Hamiltonian, (ii) an unknown symplectic (i.e. constant, nondegenerate, and skew-symmetric) structure matrix for which we need to estimate the Hamiltonian and the structure matrix, and (iii) an unknown general affine or Lie-Poisson system for which we estimate the Hamiltonian and structure constants.

3.1 Learning a Hamiltonian system with known Poisson structure

For a Hamiltonian system with a known Poisson tensor $B(x)$ the loss function (12) is given as a quadratic equation for the unknown coefficients $\alpha \in \mathbb{R}^K$ determining the unknown Hamiltonian (14). Introducing the Poisson tensor evaluated at sample points $B^{(\mu)} = B(X^{(\mu)})$ we solve the following minimization problem

$$\arg \min_{\alpha} \sum_{\mu=1}^M \|B^{(\mu)} W_{\text{in}}^{\top} \text{diag}(\alpha) \psi^{(\mu)} - F^{(\mu)}\|_F^2 + \beta \|\alpha\|_2^2. \quad (17)$$

The regularization term penalizing large norms of α serves two related purposes: first it renders a potentially ill-conditioned problem solvable, and second it prevents the occurrence of large values of entries α_{κ} which would cause the approximate Hamiltonian vector field to be stiff, requiring undesirable small time steps for its integration.

Introducing the ridge-regularized Gram matrix

$$G = \sum_{\mu=1}^M (W_{\text{in}} B^{(\mu)\top} B^{(\mu)} W_{\text{in}}^{\top}) \odot (\psi^{(\mu)} \psi^{(\mu)\top}) + \beta I_K \in \mathbb{R}^{K \times K}, \quad (18)$$

where $\odot$ denotes the Hadamard product, and

$$h = \sum_{\mu=1}^M (W_{\text{in}} B^{(\mu)\top} F^{(\mu)}) \odot \psi^{(\mu)} \in \mathbb{R}^K, \quad (19)$$

the normal equation with respect to α of the minimization problem (17) is given by

$$G\alpha = h, \quad (20)$$

with solution given by

$$\alpha = G^{\dagger} h, \quad (21)$$

where $G^{\dagger}$ denotes the pseudoinverse of G .

In the case when $B(x) = J$ is constant, we obtain

$$G = (W_{\text{in}} J^{\top} J W_{\text{in}}^{\top}) \odot (\Psi \Psi^{\top}) + \beta I_K \in \mathbb{R}^{K \times K}, \quad (22)$$

where we used that $\sum_{\mu=1}^M \psi^{(\mu)} \psi^{(\mu)\top} = \Psi \Psi^{\top}$, and

$$h = \text{diag}(W_{\text{in}} J^{\top} F \Psi^{\top}) \in \mathbb{R}^K. \quad (23)$$

This includes the case of a canonical Hamiltonian system, J given by (3).

In our numerical implementation we do not compute the normal equation (20), but instead work directly with the overconstrained equation for α , computing the singular value decomposition of the associated design matrix Ψ and including the ridge parameter in the pseudoinverse. The solution is mathematically equivalent to (21).

3.2 Learning a Hamiltonian system with unknown constant nondegenerate structure matrix

If our goal is to infer a general Hamiltonian system from vector field samples, it is unlikely that our measurements will be conveniently grouped into canonically conjugate pairs. Under the assumption that the symplectic structure is nondegenerate (which requires d to be even), there exists a linear transformation to canonical form and the approach of the previous section may be used. Alternatively, we may infer a symplectic structure matrix J with $\Gamma = J^{-1}$ together with the Hamiltonian by solving the least squares problem

$$\arg \min_{\alpha, \hat{\Gamma}} \|W_{\text{in}}^\top \text{diag}(\alpha)\Psi - \hat{\Gamma}F\|_F^2 + \beta\|\alpha\|_2^2. \quad (24)$$

under the constraint that $\hat{\Gamma}$ is skew-symmetric. The gradient of the loss function is linear in both α and $\hat{\Gamma}$. The trivial minimizer $\hat{\Gamma} = 0$, $\alpha = 0$ reflects the scaling symmetry of the Hamiltonian system: $H \mapsto \xi H$, $J \mapsto \xi^{-1}J$. To obtain a unique solution we require without loss of generality that $\|\hat{\Gamma}\|_F^2 = 1$.

Since $\Gamma = -\Gamma^\top$ there are only $q = \binom{d}{2}$ distinct entries of Γ . Let $r = r(i, j)$, $i < j$ denote an ordering, choose a basis $S_1, \dots, S_q$ of the space of skew-symmetric matrices with

$$S_{r(i,j)} = \frac{1}{\sqrt{2}}(e_i e_j^\top - e_j e_i^\top) \quad i < j, \quad (25)$$

and write $\hat{\Gamma} = \sum_{r=1}^q \gamma_r S_r$ with $\gamma \in \mathbb{R}^q$. Note the S_r are scaled such that $\|\gamma\|_2^2 = 1$ implies $\|\Gamma\|_F^2 = 1$. Substituting into the data mismatch term of the loss function (24), we obtain

$$\|W_{\text{in}}^\top \text{diag}(\alpha)\Psi - \hat{\Gamma}F\|_F^2 = \left\| \sum_{\kappa=1}^K \alpha_\kappa w_\kappa \psi_\kappa^\top - \sum_{r=1}^q \gamma_r S_r F \right\|_F^2. \quad (26)$$

The normal equation for α , analogous to (20), is

$$\tilde{G}_\beta \alpha = \tilde{h}, \quad (27)$$

where the Gram matrix in this case is given by

$$\tilde{G}_\beta = \sum_{\mu=1}^M (W_{\text{in}} W_{\text{in}}^\top) \odot (\psi^{(\mu)} \psi^{(\mu)\top}) + \beta I_K \in \mathbb{R}^{K \times K}. \quad (28)$$

Since $\tilde{h}$ is linear in $\hat{\Gamma}$ (cf (19)), it is linear in γ . We find

$$\tilde{h}(\Gamma) = D\gamma, \quad (29)$$

for $D \in \mathbb{R}^{K \times q}$ with

$$D_{kr} = w_k^\top S_r F \psi_k, \quad (30)$$

where $\psi_k \in \mathbb{R}^M$ is the transpose of the k -th row of Ψ .

Since $D\gamma$ is in the range of $\tilde{G}_\beta$ we can write the minimizer of (24) with respect to α as

$$\alpha = \tilde{G}_\beta^\dagger D\gamma. \quad (31)$$

Since $\alpha^\top \tilde{G}_\beta \alpha = \alpha^\top D\gamma$ for the minimizer α , the data mismatch loss function, evaluated at the minimizer, becomes

$$\begin{aligned} \mathcal{L}(\gamma) &= \|W_{\text{in}}^\top \text{diag}(\alpha) \Psi - \hat{\Gamma} F\|_F^2 \\ &= \gamma^\top H \gamma - 2\alpha^\top D\gamma + \alpha^\top \tilde{G}_\beta \alpha \\ &= \gamma^\top H \gamma - \alpha^\top D\gamma \\ &= \gamma^\top (H - D^\top \tilde{G}_\beta^\dagger D) \gamma, \end{aligned} \quad (32)$$

where we introduced $H \in \mathbb{R}^{q \times q}$ with

$$H_{rs} = \text{tr}(S_r F F^\top S_s^\top), \quad (33)$$

and where we used that $\tilde{G}_\beta^\dagger$ is symmetric.

Hence the minimum is given as the eigenvector corresponding to the smallest eigenvalue λ of

$$(H - D^\top \tilde{G}_\beta^\dagger D) \gamma = \lambda \gamma, \quad (34)$$

with $\|\gamma\|_2^2 = 1$.

In our numerical implementation we solve (34) for γ , construct $\hat{J} = \hat{\Gamma}^{-1}$ and rescale such that $\max_{ij} |\hat{J}_{ij}| = 1$. Thereafter we apply OMP to reduce the feature pool. Subsequently we solve for α using the methodology of Section 3.1.

3.3 Learning a Hamiltonian system with Lie-Poisson or affine Poisson structure

In coordinates in $\mathbb{R}^d$ the Lie-Poisson structure can be considered a special case ($J = 0$) of the affine Poisson structure (8). In this section we consider the inference of the more general affine Poisson structure (8) along with the Hamiltonian.

The unknown Poisson tensor $B(x)$ defined in (8) is linear in the structure coefficients C_{ij}^k and J_{ij} , which in turn are subject to the constraints imposed by the Jacobi identity (7) and the co-cycle condition (9). We parameterize $\hat{B}$ by its distinct coefficients up to skew-symmetry, denoting $\theta = (\hat{C}_{ij}^k, \hat{J}_{ij})$. We define the loss function

$$\mathcal{L}(\alpha, \theta) = \sum_{\mu=1}^M \|\hat{B}(X^{(\mu)}) \nabla \hat{H}(X^{(\mu)}) - F^{(\mu)}\|_F^2, \quad (35)$$

where $\hat{H}$ is linear in the coefficients α , and $\hat{B}$ is linear in θ subject to constraints $g(\theta) = 0$ that encode (7) and (9).

To solve for α and θ we iterate, constructing the sequence $(\theta^{(n)}, \alpha^{(n)})$ by first minimizing $\mathcal{L}$ with respect to θ for fixed α subject to (7) and (9), then solving for α by minimizing (17) with explicit solution (21) as described in Section 3.1. This process of bilevel optimization is repeated until convergence. To start the iteration we require an initial guess $\alpha^{(0)}$, which we construct using the approach of the following subsection.

3.3.1 Initialisation step: Estimating a first integral from vector field data

We adapt the methodology from [44]. A smooth first integral $Z(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ of (1) satisfies $\nabla Z(x)^\top f(x) = 0$ for all x . However, any such function is nonunique, and if there are multiple first integrals $Z_1, \dots, Z_r$, then $q(Z_1, \dots, Z_r)$ is also a first integral for any smooth q . Moreover, $Z(x) = \text{const}$ is trivially a constant of motion.

We normalize the vector field data, defining $\tilde{F}^{(\mu)} = F^{(\mu)} / \|F^{(\mu)}\|$. To find a non-trivial first integral we express it as a linear combination of random features analogously to (14) with coefficients $\alpha^{(0)}$, and we seek $\alpha^{(0)}$ minimizing the constrained loss function

$$\mathcal{L}_{\text{init}}(\alpha, \lambda) = \sum_{\mu=1}^M \|\nabla \hat{Z}(X^{(\mu)})^\top \tilde{F}^{(\mu)}\|_2^2 - \lambda \left(\sum_{\mu=1}^M (\|\nabla \hat{Z}(X^{(\mu)})\|_2^2) - 1 \right), \quad (36)$$

where $\hat{Z}$ is linear in $\alpha^{(0)}$ and λ is a Lagrange multiplier for the constraint, which precludes trivial constant first integrals with $\nabla \hat{Z}(x) \equiv 0$. We rewrite the loss function explicitly as

$$\mathcal{L}_{\text{init}}(\alpha, \lambda) = \sum_{\mu=1}^M \|\tilde{F}^{(\mu)\top} W_{\text{in}}^\top \text{diag}(\psi^{(\mu)}) \alpha\|_F^2 - \lambda \left(\sum_{\mu=1}^M \|W_{\text{in}}^\top \text{diag}(\psi^{(\mu)}) \alpha\|_2^2 - 1 \right). \quad (37)$$

Introducing

$$\begin{aligned} \tilde{A} &= \sum_{\mu=1}^M \text{diag}(\psi^{(\mu)}) W_{\text{in}} \tilde{F}^{(\mu)} \tilde{F}^{(\mu)\top} W_{\text{in}}^\top \text{diag}(\psi^{(\mu)}) \\ &= [\Psi \odot (W_{\text{in}} \tilde{F})][\Psi \odot (W_{\text{in}} \tilde{F})]^\top \in \mathbb{R}^{K \times K}, \end{aligned} \quad (38)$$

the loss function reads

$$\mathcal{L}_{\text{init}}(\alpha; \lambda) = \alpha^\top \tilde{A} \alpha - \lambda(\alpha^\top \tilde{G}_0 \alpha - 1), \quad (39)$$

where $\tilde{G}_0$ is defined by (28) with $\beta = 0$. The normal equation with respect to $\alpha^{(0)}$ is given by

$$\tilde{A} \alpha^{(0)} = \lambda \tilde{G}_0 \alpha^{(0)}, \quad (40)$$

and minimization of $\mathcal{L}_{\text{init}}$ amounts to finding the generalized eigenvector corresponding to the smallest generalized eigenvalue λ .

In general $\tilde{G}_0$ is not invertible. We decompose $\tilde{G}_0 = U_r \Sigma_r U_r^\top$ such that Σ_r contains all $r \leq K$ nonzero eigenvalues and $U_r \in \mathbb{R}^{K \times r}$ contains the first r eigenvectors of $\tilde{G}_0$, i.e. $U_r^\top U_r = I_r$. We solve for $\alpha^{(0)}$ in the column space of U_r . Substituting $\alpha^{(0)} = U_r \beta$ for $\beta \in \mathbb{R}^r$, the generalized eigenvalue problem (40) becomes

$$\tilde{A} U_r \beta = \lambda U_r \Sigma_r \beta. \quad (41)$$

Introducing $\tilde{\beta} = \Sigma_r^{-\frac{1}{2}} \beta$ and symmetrizing, we obtain

$$\Sigma_r^{-\frac{1}{2}} U_r^\top \tilde{A} U_r \Sigma_r^{-\frac{1}{2}} \tilde{\beta} = \lambda \tilde{\beta}. \quad (42)$$

This allows us to stably solve for the eigenvector $\tilde{\beta}_{\min}$ corresponding to the smallest eigenvalue $\lambda = \lambda_{\min}$ of a symmetric matrix. Transforming back into original coordinates, we finally obtain

$$\alpha^{(0)} = U_r \Sigma_r^{-\frac{1}{2}} \tilde{\beta}_{\min}, \quad (43)$$

which we use to construct the initial guess for the Hamiltonian (14).

This procedure to find an initial guess of the Hamiltonian finds an approximation to a first integral that in general will be a function of the Hamiltonian as well as any Casimirs and or Noether momenta present. Whereas Casimirs are irrelevant since they vanish in the Poisson bracket, the presence of Noether momenta could conceivably degrade the quality of the first guess. Nevertheless, the iteration of the following section may still converge.

3.3.2 Nonlinear least squares for structure constants

In the n th outer iteration of our bilevel iteration process, given an estimate for $\alpha^{(n-1)}$ we minimize (35) for the structure constants $\theta = (\hat{C}_{ij}^k, \hat{J}_{ij})$ of $\hat{B}(x)$, subject to constraints imposed by the Jacobi identity (7) and co-cycle condition (9), encoded in the condition $g(\theta) = 0$. In this section we propose a method for computing $\theta^{(n)}$ given $\alpha^{(n-1)}$.

We emphasize that this problem is challenging. The number of distinct (up to skew-symmetry) coefficients C_{ij}^k is $d \cdot \binom{d}{2}$ and of J_{ij} is $\binom{d}{2}$, i.e., the combined set of parameters

θ grows as $\mathcal{O}(d^3)$. The number of distinct constraints imposed by the Jacobi identity (7) is $d \cdot \binom{d}{3}$ and by the cocycle condition (9) is $\binom{d}{3}$, i.e., the number of constraints grows as $\mathcal{O}(d^4)$. The number of constraints precisely equals the number of parameters for $d = 5$. Consequently, for $d > 5$ an increasingly large number of constraints must converge when we find a solution. The algebraic variety $g(\theta) = 0$ defined by the constraints is reducible and stratified with singular connections. This nonsmoothness presents an obstacle to methods that rely on gradient minimization with projection onto the variety. Instead we employ a weak constraint with a continuation strategy in the constraint parameter λ . Starting from a point off the variety that simply minimizes the data mismatch, we approach the variety by gradually increasing λ until the Jacobi identity and cocycle conditions are satisfied.

For fixed $\alpha = \alpha^{(n-1)}$ we formulate (35) abstractly as a weak-constraint nonlinear least squares problem for θ

$$\arg \min_{\theta} \mathcal{L}(\theta, \alpha^{(n-1)}) + \lambda \|g(\theta)\|_2^2. \quad (44)$$

where λ is a weak constraint parameter.

We solve (44) using continuation over λ , where for small λ we fit the data well and as λ increases we enforce the constraint. Specifically, we solve (44) using an inner iteration over a schedule of λ values of increasing magnitude $\lambda_j = 10^{3j-4}$, $j = 1, \dots, j_{\max}$.

For fixed λ_j , $j > 0$ we compute $\vartheta^{(j)}$ from (44) using Levenberg-Marquardt iterations applied to the residual

$$r(\theta) = D\mathcal{L}(\theta) + \lambda g(\theta) Dg(\theta) = 0$$

from a starting value $\vartheta^{(j-1)}$, the output from the previous iteration computed with λ_{j-1} . The Levenberg-Marquardt iteration requires the Hessian of g , which is a constant matrix.

Let $\vartheta = h_{\text{LM}}(\vartheta_0, \lambda; \alpha)$ denote the solution of the Levenberg-Marquardt iteration with starting value ϑ_0 , weak constraint parameter λ , and fixed α . Then we solve the schedule

$$\vartheta^{(j)} = h_{\text{LM}}(\vartheta^{(j-1)}, \lambda_j; \alpha^{(n-1)}), \quad j = 1, \dots, j_{\max},$$

with $\vartheta^{(0)}$ given by solving the unconstrained (linear) least squares problem for the parameters. Having completed the full schedule, we define $\theta^{(n)} = \vartheta^{(j_{\max})}$. In all numerical experiments in the following Sections we set $j_{\max} = 8$.

Given $\theta^{(n)}$ we compute $\alpha^{(n)}$ by minimizing (20) using the methodology of Section 3.1.

We summarize the algorithm in the following pseudocode:

Algorithm 1 Method for jointly learning Hamiltonian and affine Poisson structure constants $\theta = (J_{ij}, C_{ij}^k)$

```

1: Solve (43) for  $\alpha^{(0)}$ 
2: for  $n = 1, \dots$  do
3:    $\vartheta^{(0)} \leftarrow \arg \min_{\theta} \mathcal{L}_3(\theta)$  from (44) with  $\lambda = 0$ .
4:   for  $j = 1, \dots, j_{\max}$  do
5:      $\lambda_j \leftarrow 10^{3j-4}$ 
6:      $\vartheta^{(j)} \leftarrow h_{\text{LM}}(\vartheta^{(j-1)}, \lambda_j; \alpha^{(n-1)})$ 
7:   end for
8:    $\theta^{(n)} \leftarrow \vartheta^{(j_{\max})}$ 
9:   Solve (20) for  $\alpha^{(n)}$ 
10: end for

```

4 Numerical integration of the inferred model via a Poisson splitting

Having inferred a Hamiltonian or Lie-Poisson vector field from sample data, one may wish to make predictions by numerically integrating the inferred system. It is generally advised to use numerical methods that are themselves symplectic or Poisson maps when integrating such systems [41, 14, 24]. One class of numerical schemes that are symplectic for general constant, skew-symmetric Poisson tensors is the Gauss-Legendre family of Runge-Kutta methods. However, they have the drawback of being implicit, requiring computationally costly root solvers. Alternatively, splitting methods lead to explicitly computable methods that are applicable to both symplectic and Poisson flows.

We begin with symplectic Hamiltonian systems. Because the exact flow of a Hamiltonian system is a symplectic map (with respect to the particular structure matrix J (or its learned approximation $\hat{J}$)), and the composition of symplectic maps is again a symplectic map, we split the Hamiltonian as a sum of individual random feature maps:

$$\hat{H}(x) = \sum_{\kappa=1}^K \hat{H}_{\kappa}(x), \quad \hat{H}_{\kappa}(x) = \alpha_{\kappa} \tanh(w_{\kappa}^{\top} x + b_{\kappa}). \quad (45)$$

The flow generated by the lone feature $\hat{H}_{\kappa}$ is the solution of

$$\dot{x} = J \nabla \hat{H}_{\kappa}(x) = \alpha_{\kappa} \text{sech}^2(w_{\kappa}^{\top} x + b_{\kappa}) J w_{\kappa} \quad (46)$$

Due to the skew symmetry we see that

$$w_{\kappa}^{\top} \dot{x} = \alpha_{\kappa} \text{sech}^2(w_{\kappa}^{\top} x + b_{\kappa}) w_{\kappa}^{\top} J w_{\kappa} = 0.$$

Consequently, $w_{\kappa}^{\top} x$ is a constant of motion of this flow, and the right side of (46) is constant. Therefore, the (exact) flow map associated with each Hamiltonian $\hat{H}_{\kappa}$ is simply given by

$$\varphi_t^{(\kappa)}(x_0) = x_0 + t \alpha_{\kappa} \text{sech}^2(w_{\kappa}^{\top} x_0 + b_{\kappa}) J w_{\kappa}.$$

The Lie-Trotter composition

$$\phi_{\Delta t}(x) = \varphi_{\Delta t}^{(1)} \circ \cdots \circ \varphi_{\Delta t}^{(K)}(x) \quad (47)$$

defines a first order symplectic numerical integrator. Its adjoint is the composition in reversed order

$$\phi_{\Delta t}^*(x) = \varphi_{\Delta t}^{(K)} \circ \cdots \circ \varphi_{\Delta t}^{(1)}(x).$$

For our numerical computations we use the symmetric Strang composition

$$\Phi_{\Delta t}(x) = \phi_{\Delta t/2} \circ \phi_{\Delta t/2}^*(x), \quad (48)$$

which is symplectic and second order accurate [14].

Note that the cost of the first order splitting (47) is essentially the same as the evaluation of the whole vector field, and the symmetric splitting is twice as expensive. Consequently, these methods have comparable costs to first and second order explicit Runge-Kutta methods applied to the learned system.

This splitting scheme generalizes recursively to larger K . For large K the splitting errors may become significant, typically requiring a smaller stepsize than is needed to integrate the true Hamiltonian in some examples (see below).

For inferred affine and Lie-Poisson systems, the arguments are similar. The κ th term in (45) generates the dynamics

$$\dot{x} = B(x) \nabla \hat{H}_\kappa(x) = \alpha_\kappa \operatorname{sech}^2(w_\kappa^\top x + b_\kappa) B(x) w_\kappa.$$

Again we find $w_\kappa^\top \dot{x} = 0$ by skew-symmetry. Recalling the definition (15) with $\psi_\kappa(x) = \operatorname{sech}^2(w_\kappa^\top x + b_\kappa)$ and denoting by $Q_\kappa = C(\cdot, w_\kappa, \cdot)$ the matrix obtained by contracting the structure tensor with w_κ along its second index and $y_\kappa = Jw_\kappa$, we can write

$$\dot{x} = \alpha_\kappa \psi_\kappa(x) (Q_\kappa x + y_\kappa),$$

where $\psi_\kappa(x)$ is constant along the flow. Consequently, the flow generated by $\hat{H}_\kappa$ is given by the variation of constants formula, formally as

$$\varphi_t^{(\kappa)}(x_0) = e^{t\alpha_\kappa \psi_\kappa(x_0) Q_\kappa} x_0 + Q_\kappa^{-1} (I - e^{t\alpha_\kappa \psi_\kappa(x_0) Q_\kappa}) y_\kappa. \quad (49)$$

which can be implemented by computing a matrix exponential (see [14] for efficient implementation, invertibility of Q_κ is not required). The resulting Lie-Trotter (47) and Strang (48) splittings, defined identically, yield Poisson integrators of first and second order, respectively. Of course, computing $2K$ matrix exponentials per time step may become costly.

5 Numerical results

We shall now show that the methodology of system identification using random feature maps is capable of reliably inferring Hamiltonian dynamical systems and their underlying symplectic, Lie-Poisson or affine Poisson structure. We consider two models: The Lorenz 86 model which comes in two flavours; a 5-dimensional version with a nontrivial affine Poisson structure and a 4-dimensional symplectic version. We further consider the 6-dimensional Kirchhoff equations which are a Lie-Poisson system.

We will employ the methods described in the previous sections to learn the Hamiltonian and/or the underlying structure matrix. In all computations we assume noise-free vector field data. The internal weights for the random feature maps, W_{in} and b_{in} are sampled using the hit-and-run algorithm [28].

5.1 The models

We first introduce the models and define the Poincaré section which will be used to assess the accuracy of the learned models.

5.1.1 The Lorenz 1986 model

In [26] E.N. Lorenz constructed a low-dimensional model based on the rotating shallow water equations to study simplified dynamics of geostrophic balance. The model equations (rescaled after [4]) are

$$\dot{x}_1 = -x_2x_3 + bx_2x_5, \quad (50a)$$

$$\dot{x}_2 = x_1x_3 - bx_1x_5, \quad (50b)$$

$$\dot{x}_3 = -x_1x_2, \quad (50c)$$

$$\dot{x}_4 = -\frac{1}{\varepsilon}x_5, \quad (50d)$$

$$\dot{x}_5 = \frac{1}{\varepsilon}x_4 + bx_1x_2, \quad (50e)$$

where b represents the rotational Froude number and $\varepsilon = b\rho(1+b^2)^{1/2}$ is a perturbation parameter (here ρ is the Rossby number; we assume $\rho = 1$). The system (50), in the following referred to as the L86 system, is a Poisson system with affine Poisson structure (8), with nonzero elements

$$\{C_{13}^2 = -1, C_{15}^2 = b, C_{23}^1 = 1, C_{25}^1 = -b, \}, \quad J_{45} = -1/\varepsilon, \quad (51)$$

up to skew-symmetry, and with Hamiltonian

$$H(x) = \frac{1}{2}(x_1^2 + 2x_2^2 + x_3^2 + x_4^2 + x_5^2). \quad (52)$$

In addition the L86 system (50) preserves the Casimir

$$Z(x) = \frac{1}{2}(x_1^2 + x_2^2). \quad (53)$$

By restricting to a level set of constant Casimir Z , Bokhove and Shepherd [4] expressed the 5-dimensional L86 model as a canonical symplectic 4-dimensional model given by

$$\dot{q}_1 = \varepsilon p_1, \quad (54a)$$

$$\dot{q}_2 = p_2, \quad (54b)$$

$$\dot{p}_1 = -\varepsilon Z \sin 2(q_1 + b\sqrt{\varepsilon}q_2), \quad (54c)$$

$$\dot{p}_2 = -q_2 - \varepsilon^{3/2}bZ \sin 2(q_1 + b\sqrt{\varepsilon}q_2). \quad (54d)$$

The canonical coordinates are defined with respect to Lorenz's formulation by

$$q_1 = \phi - \varepsilon b x_4, \quad p_1 = x_3, \quad q_2 = \sqrt{\varepsilon} x_4, \quad p_2 = -\sqrt{\varepsilon} x_5,$$

where ϕ is the phase with respect to the Casimir Z , i.e. $x_1 = \sqrt{2Z} \cos \phi$, $x_2 = \sqrt{2Z} \sin \phi$. The canonical Hamiltonian of the reduced system is given by

$$H_{\text{canon}}(q_1, q_2, p_1, p_2) = \varepsilon H = \frac{\varepsilon}{2} [p_1^2 - Z \cos 2(q_1 + b\sqrt{\varepsilon}q_2)] + \frac{1}{2}(p_2^2 + q_2^2). \quad (55)$$

We will use this 4-dimensional representation of the L86 model in the following.

As a test problem for the methods, we compute a Poincaré section for the parameter choice

$$H = b = \frac{1}{2}, \quad \rho = Z = 1,$$

which supports both invariant tori and chaotic dynamics. We choose a Poincaré section corresponding to the (ϕ, x_3) plane at $x_4 = 0$ and $\dot{x}_4 > 0$ (in canonical coordinates, the (q_1, p_1) plane at $q_2 = 0, p_2 > 0$).

Computing the Poincaré section is a challenging test problem because it requires a sufficiently accurate approximation of the Hamiltonian and symplectic/Poisson structure to allow for an accurate long time integration. Significant errors are immediately visible in the form of distortions or topological changes to the invariant tori.

The Poincaré sections are computed using simulations to time $T = 2000$ (canonical system time scale) from a set of 50 initial conditions along the two rays:

$$p_1 = 0, \quad q_1 \in \{j\Delta q, j = 1, \dots, 25\}, \quad \Delta q = \frac{\pi}{54},$$

and

$$q_1 = 0, \quad p_1 \in \{-0.3 + j\Delta p, j = 1, \dots, 25\}, \quad \Delta p = \frac{0.3 + 1/\sqrt{2}}{27}.$$

Figure 1 shows the Poincare section computed in canonical coordinates using the Störmer-Verlet method for the canonical L86 system (54).

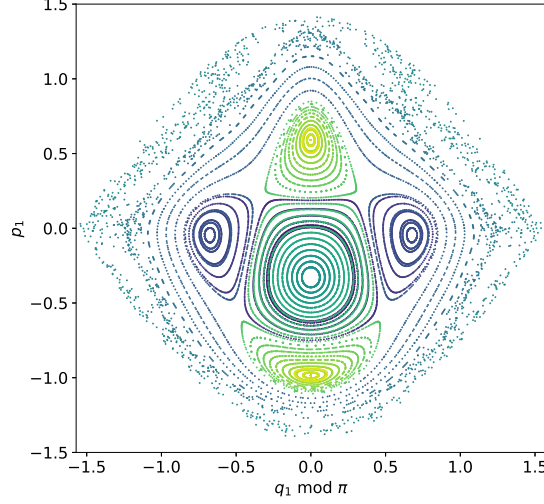


Figure 1: Poincaré section of the exact L86 model (54) computed with the Störmer-Verlet method using a stepsize $\Delta t = 0.0025$.

5.1.2 The Kirchhoff equations

We further consider a Kirchhoff-type model for rigid body motion in an ideal fluid, defined on $\mathbb{R}^6$.

$$\dot{\Omega} = -\Omega \times \frac{\partial H}{\partial \Omega} - \Gamma \times \frac{\partial H}{\partial \Gamma}, \quad \dot{\Gamma} = -\Gamma \times \frac{\partial H}{\partial \Omega}, \quad (56)$$

where $\Omega(t), \Gamma(t) \in \mathbb{R}^3$ are the angular and linear velocities, and the Hamiltonian is defined by

$$H(\Omega, \Gamma) = \frac{1}{2} \sum_{i=1}^3 a_i \Omega_i^2 + \sum_{i=1}^3 b_i \Omega_i \Gamma_i + \frac{1}{2} \sum_{i=1}^3 d_i \Gamma_i^2$$

The system is Lie-Poisson and admits two Casimir functions

$$Z_1 = \|\Gamma\|^2, \quad Z_2 = \Omega \cdot \Gamma. \quad (57)$$

5.2 Inferring the Hamiltonian for a given canonical structure

We first study inference of the Hamiltonian assuming that the canonical symplectic structure is known. We consider here the canonical L86 model (54). Given vector field data we learn the canonical Hamiltonian H_{canon} (55) by minimizing (17), as described in Section 3.1.

We generate $M = 6000$ uniform random vector field samples $(X^{(\mu)}, F^{(\mu)})_{\mu=1}^M$ in the box $(q_1, q_2, p_1, p_2) \in [-1.8, 1.8] \times [-2, 2] \times [-1.2, 1.2] \times [-1.2, 1.2]$. We generate K_{pool} random feature maps and reduce this to K using OMP.

Our method contains two hyperparameters which require tuning: K_{pool} and β . We now investigate to what extent they influence the performance. We conducted a parameter study, choosing

$$K_{\text{pool}} \in \{1000, 2000, 4000, 8000, 12000, 16000, 20000\}, \quad \log_{10} \beta \in [-9, -4],$$

and compared the relative error between the true and the learned vector fields

$$\mathcal{E}_F = \frac{\|JW_{\text{in}}^{\top} \text{diag}(\alpha)\Psi - F\|_F}{\|F\|_F} \quad (58)$$

and the learned coefficient of largest magnitude $|\alpha|_{\infty}$. The latter quantity affects the stiffness of time integration and is therefore of significance for accurate computation of Poincaré sections. For each value of K_{pool} we learn the model for several values of β and record the vector-field error $\mathcal{E}_F$ and $|\alpha|_{\infty}$. Figure 2 shows best-fit isolines of the vector-field error and of $|\alpha|_{\infty}$. The figure indicates how to choose the initial pool size K_{pool} and ridge parameter β to learn the vector field to a desired accuracy, but also provides a rough indication of how the choice affects the stiffness of the resulting learned vector field.

To reduce the number of feature maps we apply orthogonal matching pursuit (OMP) using the batch-OMP variant of [40]. Use of OMP gives good approximation error with a significantly reduced feature pool as the following table shows (fixed $\beta = 3 \times 10^{-7}$, least squares fit $K \approx 64.1K_{\text{pool}}^{0.254}$):

K_{pool}	4000	8000	12000	16000	20000
K	524	635	686	763	783

Using the exponential fit for fixed $|\alpha|_{\infty} = 150$ (i.e. along a line of approximately constant stiffness) we generated four Poincaré sections using RFM pool sizes $K_{\text{pool}} = \{2000, 4000, 8000, 16000\}$ (actual selected pool sizes $K = \{424, 524, 635, 763\}$, upper orange line in Figure 2) and (fixed) stepsize $\Delta t = 0.00125$. Figure 3 shows convergence of the Poincaré sections. As the RFM pool increases and β decreases, the Poincaré sections converge to the correct topology of invariant tori shown in Figure 1. The vector field errors (58) for these four cases are, respectively, 0.0428, 0.0181, 0.0070, 0.0021.

5.3 Inferring the Hamiltonian and the symplectic structure

In Section 3.2 we proposed a method for jointly inferring the symplectic structure matrix J and the Hamiltonian H for a system for which a canonical relationship among the variables may not be known. We consider again the canonical L86 model (54).

To test the method, we generate a random matrix L with condition number 2.6667. We generate training data using two different approaches. First, given M uniform random samples $(q_1, q_2, p_1, p_2) \in [-1.8, 1.8] \times [-2, 2] \times [-1.2, 1.2] \times [-1.4, 1.4]$, we generate transformed data $\tilde{X}^{(\mu)} = LX^{(\mu)}$ and $\tilde{F}^{(\mu)} = LF^{(\mu)}$, $\mu = 1, \dots, M$ and use this data to generate an RFM pool using the one shot hit-and-run algorithm [28].

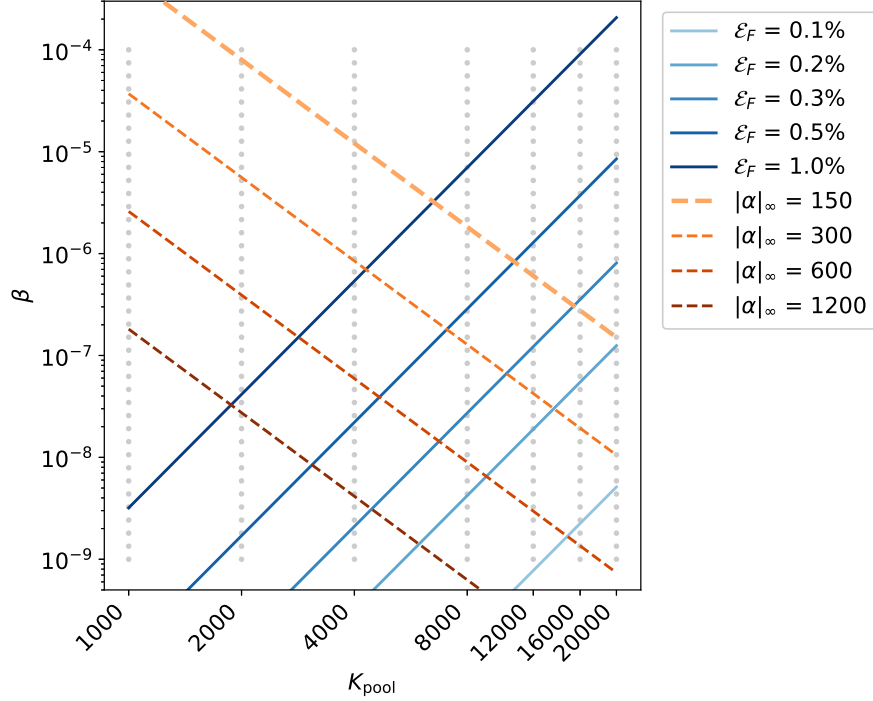


Figure 2: Effect of the hyperparameters K_{pool} and β on the accuracy of the learned Hamiltonian of the L86 model (54). We show best fit isolines of the vector field error (blue) and of the maximal learned weights $|\alpha|_{\infty}$ for various choices of RFM pool size K_{pool} and ridge parameter β . The lines show best fits obtained from results of learned models shown as points in vertical columns.

If $K \geq dM$, the matrix of the eigenvalue problem (34) is identically zero for $\beta = 0$. Even for small $\beta > 0$ the eigenproblem is overwhelmed by noise without sufficient training data. Because we solve for $\hat{J}$ before applying OMP, we require $M > K_{\text{pool}}/d$. However, a further increase in M has only a mild effect, in our experience. For $K_{\text{pool}} = 12000$ and $M = 3600$, relative error in $\hat{J}$ is 10^{-9} . Error in $\hat{J}$ decreases very rapidly, exponentially fit at a rate of about $10^{-4}K_{\text{pool}}$.

Figure 4 shows Poincaré sections of the exact system (top row) and the system with learned (transformed) symplectic structure $\hat{J}$ (bottom row), where we used $K_{\text{pool}} = 20000$ and $\beta = 5 \times 10^{-8}$. On the left, the Poincaré section is shown in the transformed coordinates of the training samples and learned dynamics and on the right after inverse transformation to canonical coordinates. The invariant tori are a slightly skewed and the topology is also not completely converged.

In Figure 5 we repeat the above experiment using alternative training data generated by computing orbits of the transformed exact system using the initial conditions that produce the Poincaré section. In this way, the training data samples precisely the part

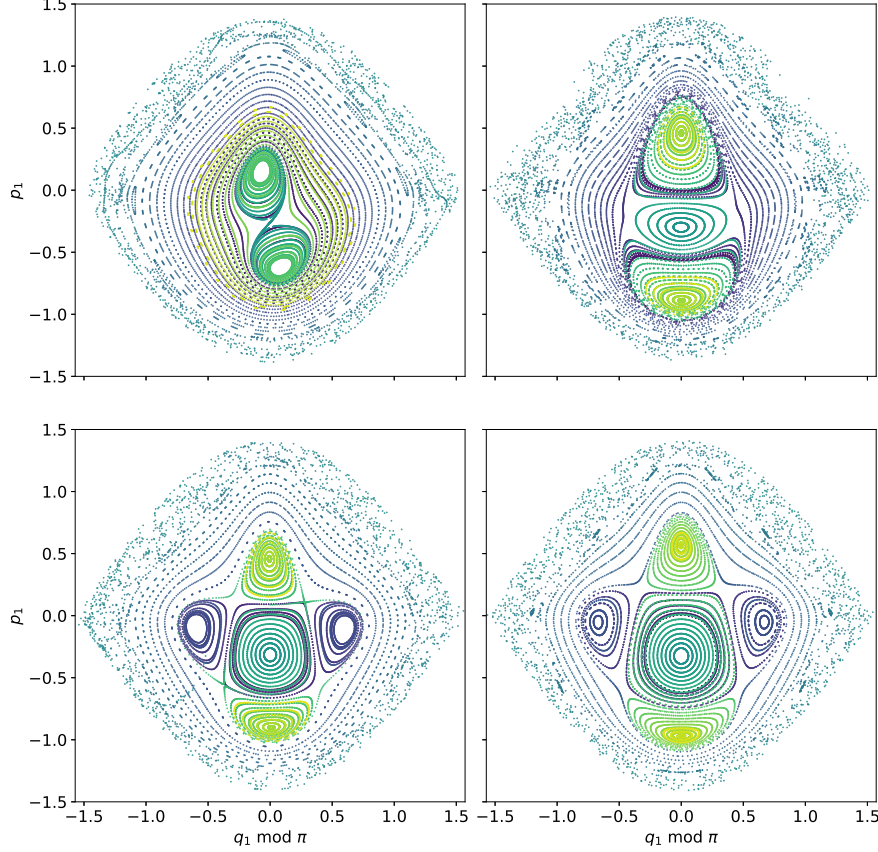


Figure 3: Convergence of Poincaré sections of the L86 model (54) in canonical coordinates with learned Hamiltonian, for a range of increasing K_{pool} , decreasing β along the isoline $|\alpha|_{\infty} = 150$ of Figure 2. (K_{pool}, β) values for upper left: $(2000, 8.18 \times 10^{-5})$, upper right: $(4000, 1.24 \times 10^{-5})$, lower left: $(8000, 1.88 \times 10^{-6})$, lower right: $(16000, 2.86 \times 10^{-7})$. We used $M = 6000$ training samples and integration stepsize $\Delta t = 0.00125$.

of phase space that includes the Poincare section. In this case the section is very well reproduced, despite a smaller initial feature pool with $K_{\text{pool}} = 12000$ and a larger stepsize $\Delta t = 0.00125$.

5.4 Inferring the Hamiltonian and an affine Poisson structure

In this section we test the full methodology of Section 3.3.2 to infer the Hamiltonian and structure constants C_{ij}^k and J_{ij} of the affine Poisson description of the L86 model (50). We choose an initial RFM pool of $K_{\text{pool}} = 500$ (no OMP), ridge parameter $\beta = 10^{-7}$, $M = 4000$ training points drawn uniformly from a rectangular box $x \in [-1.5, 1.5]^5$. The nonlinear least squares of Section 3.3.2 converges to relative vector field error 0.00036. The structure coefficients are determined to relative errors of 2.2×10^{-4} for C and to

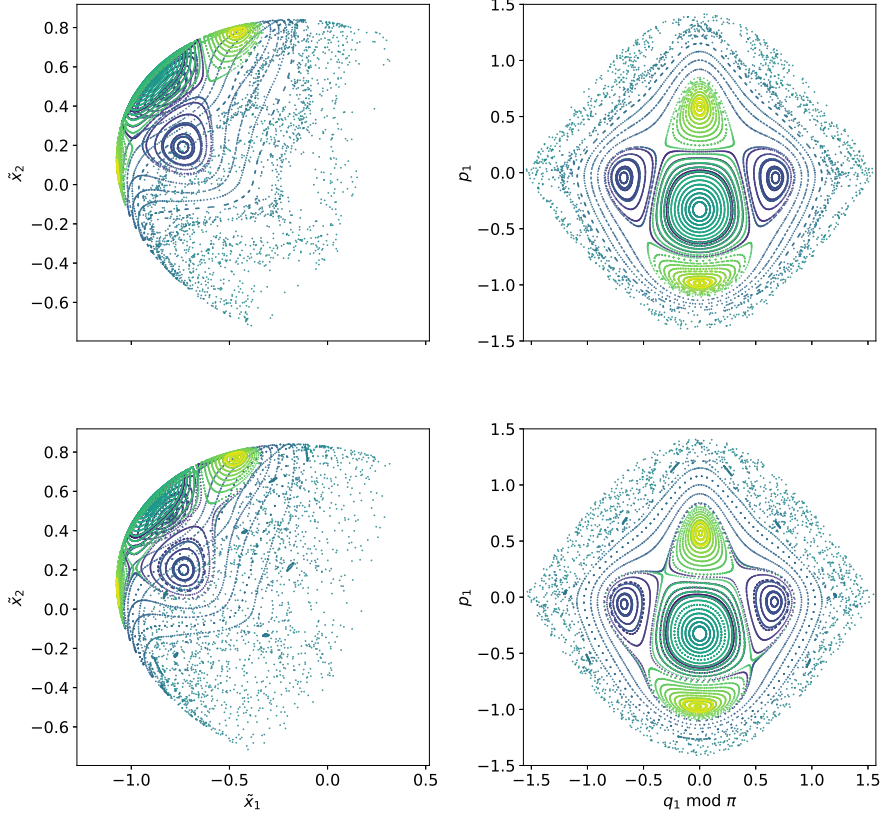


Figure 4: Poincaré sections of the L86 model (54) in transformed and in canonical coordinates. The left columns shows the Poincaré section in the transformed coordinates of the training data and learned system. The right column shows the same Poincaré sections after applying the inverse transformation. The top row shows simulations of the exact Hamiltonian using a Hamiltonian (Strang) splitting with stepsize $\Delta t = 0.0025$. The lower row shows the inferred Hamiltonian system using the splitting integrator (48) for $K_{\text{pool}} = 20000$ random feature maps, $\beta = 5 \times 10^{-8}$, $M = 6000$ training samples, and stepsize $\Delta t = 0.0003125$.

8×10^{-5} for J .

We integrate the learned model using the splitting method of Section 4. We constructed the Poincaré sections of the true system (50) using stepsize $\Delta t = 0.0025$ and the learned model using a stepsize $\Delta t = 0.02$ in the $\mathbb{R}^5$ description. Figure 6 shows the comparison, which is already very close at this relatively low resolution fit.

The Casimir of the learned model was found to be slightly perturbed with respect to the true Z . It is quadratic in x , but with a small dependence on all components and it contains an additional small linear term. Figure 7 shows the error in the Casimir and Hamiltonian with respect to the first integrals of the true system and those of the

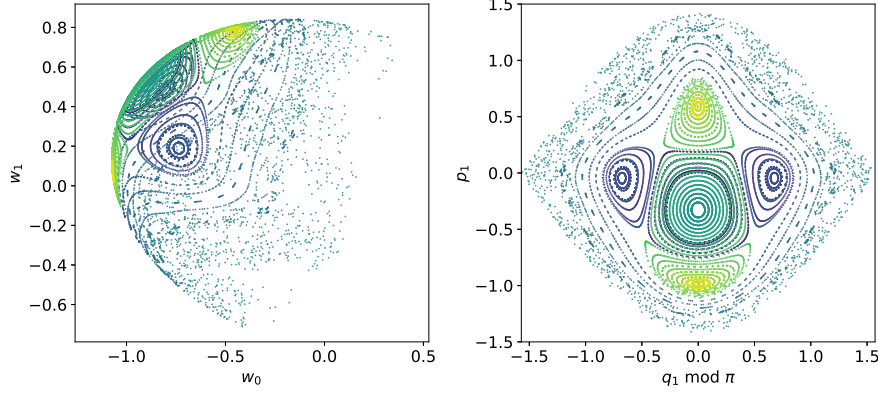


Figure 5: Poincaré sections of the learned L86 model (54) in transformed and in canonical coordinates. The training data was generated by running transformed exact model orbits with initial conditions lying on the Poincaré section. We used $K_{\text{pool}} = 12000$ random feature maps, $\beta = 5 \times 10^{-8}$, $M = 6000$ training samples and stepsize $\Delta t = 0.00125$.

learned system. The relative errors in the conserved quantities of the true system are both about 10^{-3} . The relative errors of the exact conserved quantities of the learned system are significantly smaller: about 10^{-7} for the Hamiltonian and about 10^{-12} for the Casimir. The effects of many rounding errors is evident in the slow error growth of the Casimir Z .

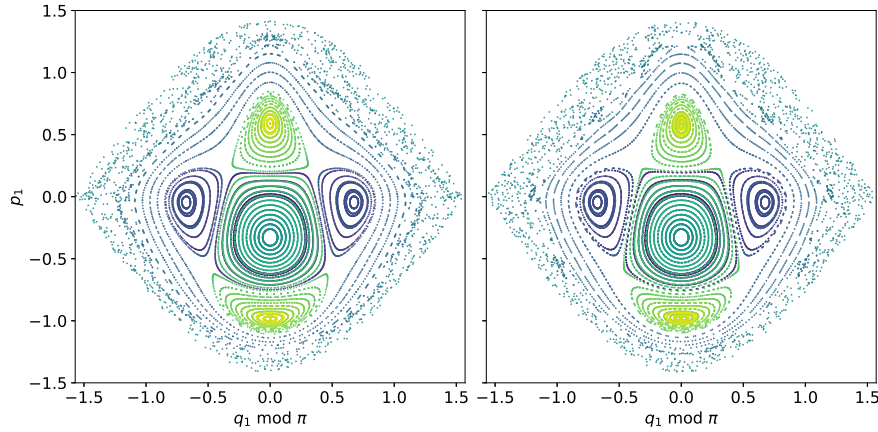


Figure 6: Poincaré section of the learned L86 model (50) on $\mathbb{R}^5$ with affine Poisson structure (51) for the exact system (left) and the learned model (right). For the true model we used a stepsize $\Delta t = 0.0025$. For the learned model we used $K = 500$ features (no OMP), $\beta = 10^{-7}$, $M = 4000$ training samples and a stepsize $\Delta t = 0.02$. Both systems were integrated using a symmetric Poisson splitting integrator.

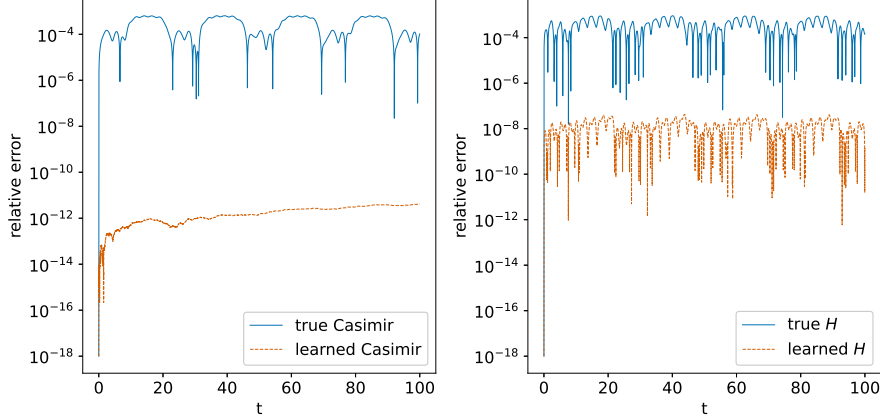


Figure 7: Relative errors in conservation laws for the learned L86 model with affine Poisson structure: (left) the error in the true system’s Casimir (53) and the exact Casimir of the learned system; (right) the error in the true system’s Hamiltonian (52) and the learned Hamiltonian.

5.5 Inferring an affine Poisson structure for a quadratic Hamiltonian

Many physical Hamiltonian systems have a quadratic Hamiltonian. As discussed in Section 2, affine Poisson (Lie-Poisson) systems remain affine Poisson (Lie-Poisson) under a linear change of coordinates. Consequently, if the Hamiltonian of the true dynamics is quadratic with $H(x) = \frac{1}{2}x^\top Mx + c^\top x$ (with $c = 0$ in the Lie-Poisson case), where M is symmetric positive definite, then there exists a linear change of variables $\tilde{x} = Lx + y$ (with $y = 0$ in the Lie-Poisson case) such that the Hamiltonian becomes isotropic in the sense that $H(\tilde{x}) = \frac{1}{2}\|\tilde{x}\|^2$. This transformation implies that without loss of generality any system with a quadratic Hamiltonian can be assumed to be of this form. Hence, the system identification problem reduces to one where only the symplectic structure matrix or the Poisson tensor needs be learned from data. This assumption significantly reduces the computational complexity and allows for higher accuracy. We illustrate this for the affine representation of the L86 model (50).

Assuming Hamiltonian $H = \frac{1}{2}\|x\|^2$, we can replace $\nabla \hat{H}(X^{(\mu)})$ with $X^{(\mu)}$ in the loss function (35), thereby eliminating the feature maps and α altogether. We solve directly for the structure parameters $\theta = (\hat{C}_{ij}^k, \hat{J}_{ij})$ using the continuation method of Section 3.3.2. We implemented this for the L86 model (50), drawing $M = 2000$ vector field data uniformly on $x \in [-1.5, 1.5]^5$. The transformed (due to isotropization) structure constants are exact to machine precision relative error. Figure 8 shows the Poincare section.

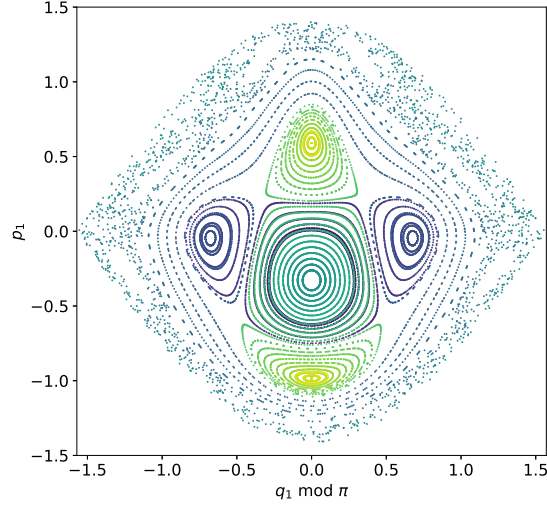


Figure 8: Poincaré section of the learned L86 model (50) for which the affine Poisson structure is inferred assuming a quadratic Hamiltonian. We used $M = 2000$ data points and a stepsize $\Delta t = 0.01$.

5.6 Inferring the Hamiltonian and the Lie-Poisson structure

We now test the methodology of Section 3.3 on Lie-Poisson systems. As stated in Section 2, the number of Jacobi identity constraints exceeds the number of structure coefficients for dimension $d > 5$. In this section we present numerical results for the Kirchhoff-type model (56) for which $d = 6$. For the numerical experiment we chose $a = (0.5452, 0.8285, 0.5628)$, $b = (-0.0756, -0.2957, 0.2476)$, and $d = (0.6759, 1.1507, 0.7016)$, and initial condition

$$\Omega(0) = (0.5, 0.3, -0.4), \quad \Gamma(0) = (1.0, 0.2, -0.3).$$

We generated $M = 6000$ phase space samples uniformly in the hypercube $x = (\Omega, \Gamma) \in [-1.5, 1.5]^6$ and learned the Lie algebra structure constants $\hat{C}_{ij}^k$ and Hamiltonian $\hat{H}$ using $K = 500$ random feature maps (without OMP) with ridge parameter $\beta = 10^{-7}$. The learned structure constants matched the true structure constants with a maximum entrywise error of 0.000132.

Figure 9 illustrates the computed orbits of the true and of the learned systems, integrated with the Poisson splitting method (48) described in Section 4 with stepsize $\Delta t = 0.01$ up to time $T = 300$. The trajectories show good agreement.

The learned system admits two Casimirs, but these may differ significantly from those of the true system due to the ambiguity of multiple Casimirs. That is, since the Poisson structure of the learned system is an approximation of the true Poisson structure, its Casimirs could be approximations of combined functions of the true system's

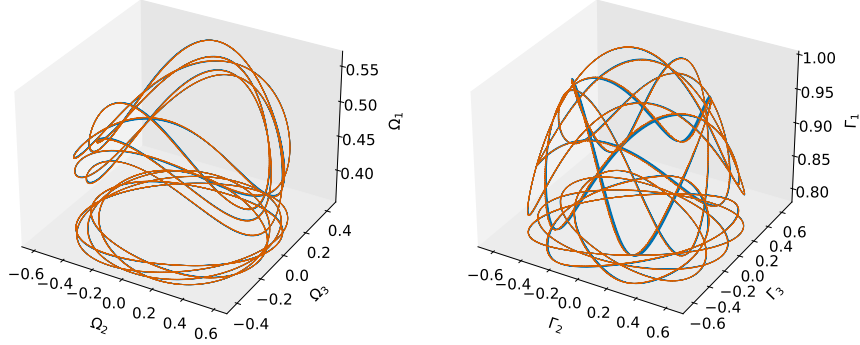


Figure 9: Trajectories of the true (blue) and learned (orange) dynamics of $\Omega(t)$ (left) and $\Gamma(t)$ (right). We used $K = 500$ features (no OMP), $\beta = 10^{-7}$, $M = 6000$ training samples, integration stepsize $\Delta t = 0.01$ and orbit length $T = 300$.

Casimirs:

$$\tilde{Z}_1 \approx g_1(Z_1, Z_2), \quad \tilde{Z}_2 \approx g_2(Z_1, Z_2).$$

Figure 10 shows errors in the Casimirs and in the Hamiltonian of the true system, evaluated along the learned system’s trajectory, as well as the error in conservation of the learned Hamiltonian and Casimirs of the learned Poisson tensor. The learned system’s Casimirs are conserved to relative error less than 10^{-10} over the whole time window. The observed drift grows at order $t^{1/2}$ consistent with random rounding errors due to the large number of terms in the splitting; we employ $2K = 1000$ matrix exponential computations per time step.

6 Discussion and outlook

We have developed a data-driven methodology for system identification of general symplectic Hamiltonian systems and for Hamiltonian systems with an affine Poisson structure, including Lie-Poisson systems. We achieved this by expressing the Hamiltonian as a random feature map. This allows for a computationally efficient estimation with an explicit solution of the least-squares regression problem in the case when the Poisson tensor is given. To learn both the Hamiltonian and the Poisson tensor we employed a bilevel optimization. The nonsmooth nature of the algebraic variety defined by the Jacobi identity for affine Poisson systems posed a particular challenge that we solved by an gradual enforcement of the constraint. We provided numerical results that show that the methodology successfully recovers the true dynamics even in systems with dimension larger than five for which the number of constraints exceeds the number of unknowns. We remark that we have done additional successful experiments with non-physical randomly generated Lie-Poisson systems with dimensions up to $\mathbb{R}^9$.

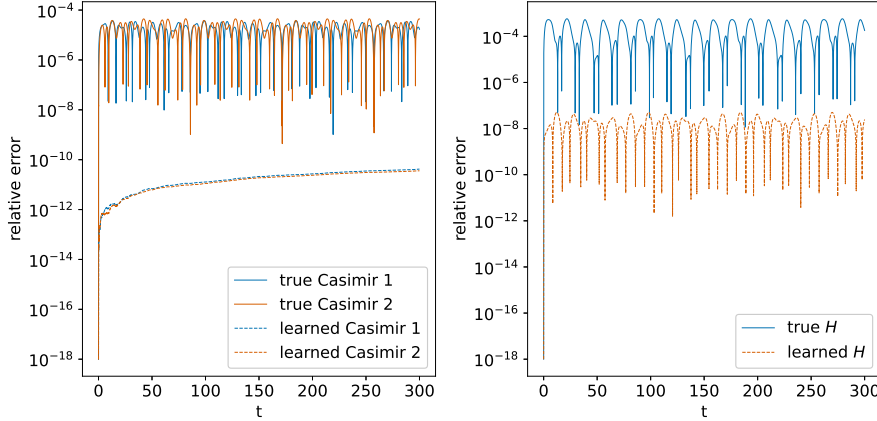


Figure 10: Error in conservation of Casimirs (left) and Hamiltonian (right) for the Kirchhoff model (56) by the learned model. Due to the ambiguity between the Hamiltonian and Casimirs, the learned model’s exact first integrals differ from those of the true model. In the figures we indicate conservation errors with respect to the true model’s first integrals and those of the learned model. Integration stepsize is $\Delta t = 0.01$.

The methodology as introduced here relies on several assumptions, that we now discuss. We assume here that we have access to the whole state variables $x \in \mathbb{R}^{2d}$ so in particular, we assume that we observe the conjugate momenta in the canonical setting. This may not be the case in a real-world situation.

Further, we have considered here the case where we are given samples of the state space variables and the vector field, i.e. the time derivative of the vector field. In practice one typically faces the situation where one has access to noisy measurements of the state space variables only. In this case, one would need to estimate the vector field from the state space variables. The presence of noise prohibits any estimation of the vector field via finite-time discretizations, even for small sampling times. One may either employ denoising via splines [20], control the noise via data assimilation [12] or by Gaussian process regression [19]. The important case of partial observations is also not considered here and may be tackled via delay embedding [11].

We chose here RFMs to express the Hamiltonian. One may instead express the Hamiltonian as a linear combination of functions from a prescribed library. This opens up the way to perform SINDy [5] and achieve a sparse identification of Lie-Poisson systems.

An interesting application of our framework would be to actually discover if a dynamical system is Lie-Poisson. For the heavy top, for example, we could take measurements in Cartesian coordinates. The associated Hamiltonian system in these coordinates is then not Lie-Poisson. However, by allowing for a diffeomorphic transformation $z = T(x)$ we can test if the dynamical system allows for a Lie-Poisson structure. We can

represent the transformation T by a random feature map and impose the constraints for the structure matrix to be Lie-Poisson as described in Section 3.3. However, the transformation is now entering the nonlinear activation function and the loss function will therefore not be a quadratic function of the coefficients of the random feature map for the transformation.

AI statement. Large language model tools (Claude, Anthropic) were used to assist with implementing the numerical experiments and figure-generation code described in this article, based on the mathematical formulation, algorithms, and experimental design specified by the authors, and to assist in checking the internal mathematical consistency of the derivations in this manuscript. A first version of the abstract, and this AI statement itself, were also generated by the model. All code, numerical results, and mathematical content were reviewed and verified by the authors, who take full responsibility for their correctness.

References

- [1] A. Barron. Universal approximation bounds for superposition of a sigmoidal function. *IEEE Trans. on Inform. Theory*, 39:930–945, 1993.
- [2] T. Bertalan, F. Dietrich, I. Meziz, and I. G. Kevrekidis. On learning Hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):121107, 12 2019.
- [3] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, 2023.
- [4] O. Bokhove and T. G. Shepherd. On Hamiltonian balanced dynamics and the slowest invariant manifold. *Journal of Atmospheric Sciences*, 53(2):276–297, 1996.
- [5] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [6] G. Cybenko. Approximation by superposition of a sigmoidal function. *Math. Contr., Sign., and Syst.*, 2:303–314, 1989.
- [7] M. David and F. Méhats. Symplectic learning for Hamiltonian neural networks. *Journal of Computational Physics*, 494:112495, 2023.
- [8] O. R. A. Dunbar, N. H. Nelsen, and M. Mutic. Hyperparameter optimization for randomized algorithms: a case study on random features. *Statistics and Computing*, 35(3):56, 2025.

- [9] M. Dyring Hansen, M. Ghirardelli, E. Celledoni, D. M. de Diego, and B. Owren. Learning forced multibody dynamics on lie groups. *arXiv e-prints*, pages arXiv–2607, 2026.
- [10] C. Eldred, F. Gay-Balmaz, S. Huraka, and V. Putkaradze. Lie–Poisson neural networks (LPNets): Data-based computing of Hamiltonian systems with symmetries. *Neural Networks*, 173:106162, 2024.
- [11] G. A. Gottwald and S. Reich. Combining machine learning and data assimilation to forecast dynamical systems from noisy partial observations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(10):101103, 10 2021.
- [12] G. A. Gottwald and S. Reich. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Physica D: Nonlinear Phenomena*, 423:132911, 2021.
- [13] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [14] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations*. Springer-Verlag, second edition, 2006.
- [15] C.-D. Han, B. Glaz, M. Haile, and Y.-C. Lai. Adaptable Hamiltonian neural networks. *Phys. Rev. Res.*, 3:023156, May 2021.
- [16] M. D. Hansen, E. Celledoni, and B. K. Tapley. Learning mechanical systems from real-world data using discrete forced lagrangian dynamics. *arXiv preprint arXiv:2505.20370*, 2025.
- [17] D. D. Holm, T. Schmäh, and C. Stoica. *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions*. Oxford University Press, Oxford, 2009.
- [18] J. Hu, J.-P. Ortega, and D. Yin. A global structure-preserving kernel method for the learning of Poisson systems. *Journal of Nonlinear Science*, 35(4):79, 2025.
- [19] J. Hu, J.-P. Ortega, and D. Yin. A structure-preserving kernel method for learning Hamiltonian systems. *Math. Comput.*, 95:1719–1774, 2025.
- [20] K. Kaheman, J. N. Kutz, and S. L. Brunton. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. A*, 476(2242):20200279, 25, 2020.
- [21] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

- [22] K. Lee, N. Trask, and P. Stinis. Machine learning structure preserving brackets for forecasting irreversible processes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5696–5707. Curran Associates, Inc., 2021.
- [23] C. Legaard, T. Schranz, G. Schweiger, J. Drgoňa, B. Falay, C. Gomes, A. Iosifidis, M. Abkar, and P. Larsen. Constructing neural network based models for simulating dynamical systems. *ACM Comput. Surv.*, 55(11), Feb. 2023.
- [24] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, 2004.
- [25] M. E. Levine and A. M. Stuart. A framework for machine learning of model error in dynamical systems. *Comm. Amer. Math. Soc.*, 2:283–344, 2022.
- [26] E. N. Lorenz. On the existence of a slow manifold. *Journal of Atmospheric Sciences*, 43(15):1547–1558, 1986.
- [27] P. Mandal and G. A. Gottwald. Learning dynamical systems with hit-and-run random feature maps. *Nature Communications*, 16(1):5961, 2025.
- [28] P. Mandal, G. A. Gottwald, and N. Cranch. On the choice of the non-trainable internal weights in random feature maps for forecasting chaotic dynamical systems. *Foundations of Data Science*, 8(0):168–199 EP –, 2026.
- [29] J. E. Marsden and T. Ratiu. *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*, volume 17 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2nd edition, 1999.
- [30] N. H. Nelsen and A. M. Stuart. The random feature model for input-output maps between Banach spaces. *SIAM Journal on Scientific Computing*, 43(5):A3212–A3243, 2021.
- [31] J. S. North, C. K. Wikle, and E. M. Schliep. A review of data-driven discovery for dynamic systems. *International Statistical Review*, 91(3):464–492, 2023.
- [32] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.
- [33] M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, editors. *Automatic-differentiated Physics-Informed Echo State Network (API-ESN)*, Cham, 2021. Springer International Publishing.
- [34] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön. Deep networks for system identification: A survey. *Automatica*, 171:111907, 2025.
- [35] I. Price, A. Sanchez-Gonzalez, F. Alet, T. R. Andersson, A. El-Kadi, D. Masters, T. Ewalds, J. Stott, S. Mohamed, P. Battaglia, R. Lam, and M. Willson. Probabilistic weather forecasting with machine learning. *Nature*, 2024.

- [36] B. Prokop and L. Gelens. Data-driven discovery of dynamical models in biology. *Nature Reviews Physics*, 2026.
- [37] V. Putkaradze. Latent Lie-Poisson neural networks (LLPNNs): Discovering the motion of Lie-Poisson systems through observable data and latent dynamics. *arXiv e-prints*, 2026.
- [38] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.
- [39] A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561, 2008.
- [40] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient impliemntation of the K-SVD algorithm and the batch-OMP method. Technical Report CS-2008-08, Technion, 2008.
- [41] J.-M. Sanz-Serna and M.-P. Calvo. *Numerical Hamiltonian Problems*. Chapman & Hall, 1994.
- [42] C. Schötz, A. White, M. Gelbrecht, and N. Boers. Machine learning for predicting chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 36(5):053105, 05 2026.
- [43] Z. Zhang, Y. Shin, and G. Em Karniadakis. GFINNs: GENERIC formalism informed neural networks for deterministic and stochastic dynamical systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229):20210207, 06 2022.
- [44] W. Zhu, H.-K. Zhang, and P. Kevrekidis. Machine learning of independent conservation laws through neural deflation. *Physical Review E*, 108(2):L022301, 2023.