Data-driven Prediction of General Hamiltonian Dynamics via Learning Exactly-Symplectic Maps

Renyi Chen [1] **Molei Tao** [1]

Abstract

We consider the learning and prediction of nonlinear time series generated by a latent symplectic map. A special case is (not necessarily separable) Hamiltonian systems, whose solution flows give such symplectic maps. For this special case, both generic approaches based on learning the vector field of the latent ODE and specialized approaches based on learning the Hamiltonian that generates the vector field exist. Our method, however, is different as it does not rely on the vector field nor assume its existence; instead, it directly learns the symplectic evolution map in discrete time. Moreover, we do so by representing the symplectic map via a generating function, which we approximate by a neural network (hence the name GFNN). This way, our approximation of the evolution map is always *exactly* symplectic. This additional geometric structure allows the local prediction error at each step to accumulate in a controlled fashion, and we will prove, under reasonable assumptions, that the global prediction error grows at most *linearly* with long prediction time, which significantly improves an otherwise exponential growth. In addition, as a map-based and thus purely data-driven method, GFNN avoids two additional sources of inaccuracies common in vector-field based approaches, namely the error in approximating the vector field by finite difference of the data, and the error in numerical integration of the vector field for making predictions. Numerical experiments further demonstrate our claims.

1. Introduction

Given a collection of sequences, each being a multidimensional time series produced by the same latent mechanism, we consider learning this mechanism and predicting a sequence's future evolution. More precisely, suppose there is an unknown (possibly highly nonlinear) map ϕ that evolves any initial condition in discrete time i according to

$$x_{i+1} = \phi(x_i).$$

Provided with training data $\{x_{i,j}\}$, where each $j \in [M]$ corresponds to such a sequence ($i \in \{0, \dots, N_j\}$), we'd like to learn ϕ purely from the data, or more precisely an approximation of its image $\tilde{\phi}(x) \approx \phi(x)$ for any x in the problem domain. This way, one can for example perform continuation of existing sequences via $\tilde{x}_{i+1,j} = \tilde{\phi}(\tilde{x}_{i,j})$ for $i \geq N_j$ and $\tilde{x}_{N_j,j} = x_{N_j,j}$, or predict a sequence evolved from a new initial condition via $y_{i+1} = \tilde{\phi}(y_i)$ for $i > 0$.

This task of course appears in many contexts (see e.g., Sec.1.1). This article considers a very specific one, for which the latent map ϕ is assumed to be *symplectic*: for simplicity we will work with finite dim. vector spaces equipped with canonical symplectic structure, which means each x can be written as $x = [\boldsymbol{p}, \boldsymbol{q}]$ where $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^d$, and the Jacobian of ϕ satisfies

$$(\phi')^T J \phi' = J,$$

where $J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$ is a $2d$-by-$2d$ matrix with 0 and I being d-by-d blocks.

The consideration of symplectic evolution maps is largely motivated by the learning and prediction of mechanical behaviors, which recently attracted significant attention (see Sec.1.1 3rd paragraph). More precisely, if the latent evolution mechanism is provided by a Hamiltonian mechanical system, each time series is given by a solution to the Hamiltonian ODE system sampled at discretized time points. That is, $x_i = [\boldsymbol{p}_i, \boldsymbol{q}_i]$, $\boldsymbol{p}_i = \boldsymbol{p}(ih)$, $\boldsymbol{q}_i = \boldsymbol{q}(ih)$, with

$$\dot{\boldsymbol{p}}(t) = -\frac{\partial H}{\partial \boldsymbol{q}}(\boldsymbol{p}(t), \boldsymbol{q}(t)),\ \dot{\boldsymbol{q}}(t) = \frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{p}(t), \boldsymbol{q}(t)), \quad (1)$$

where $H(\cdot, \cdot)$ is the latent Hamiltonian function and $h > 0$ is the sampling time step. In this case, the latent ϕ we're

[1] School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA. Correspondence to: Molei Tao <mtao@gatech.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

trying to learn is the flow map of ODE (1), defined as

$$\phi[\boldsymbol{p}(t), \boldsymbol{q}(t)] := [\boldsymbol{p}(t+h), \boldsymbol{q}(t+h)], \quad \forall t.$$

Given any $h > 0$, the corresponding ϕ is a symplectic map (e.g., Goldstein, 1980), and that is why the learning of symplectic maps is relevant.

Worth noting is, a popular and successful line of thoughts is based on learning the right hand side of the latent ODE, which in our case (eq.1) amounts to either generic approaches that learn the vector-valued function $f(\boldsymbol{p}, \boldsymbol{q}) := [-\partial H/\partial \boldsymbol{q}, \partial H/\partial \boldsymbol{p}]$ (see Sec.1.1 1st paragraph), or specialized methods that utilize the problem structure and learn the scalar-valued function $H(\boldsymbol{p}, \boldsymbol{q})$ (see Sec.1.1 3rd paragraph). This article, however, is based on a different idea, namely to directly **learn the evolution map** ϕ.

The advantages of doing so include: (i) Generality: it works no matter whether there is an underlying ODE system (see Sec.4.4 for an example where there isn't). (ii) Local Accuracy: for purely data driven problems, learning the map has *one* source of error, namely the approximation error of the map, whereas learning vector field generally has *three*: first one has to estimate the vector field from data, for example by finite difference which incurs error, then there is approximation error of the vector field, and finally the vector field needs to be numerically integrated in order to make predictions and this creates error too. (iii) Symplecticity (and Global Accuracy): we will propose a simple way to exactly maintain the symplecticity of ϕ despite using an approximation[1]; we will also rigorously show, when considering how local error accumulates for making long time predictions, exact symplecticity can help significantly by keeping the error accumulation additive, so that global error grows linearly instead of exponentially.

Example 1. *To make things concrete, consider latent Hamiltonian dynamics $\dot{x} = Ax$ where $A = -A^T$. A vector-field-based method aims at learning the function Ax, but if discrete time-series are the only available data, it actually learns $\tilde{A}x$ instead, where $\tilde{A} = (\exp(Ah) - I)/h$ if 1st-order finite difference is used for estimating the vector-field. A map-based method seeks the map $\exp(Ah)x$ instead. When later making predictions, the vector-field-based method numerically integrates the vector field $\tilde{A}x$, which oftentimes corresponds to constructing a polynomial in h approximation of $\exp(\tilde{A}h)$; on the contrary, a map-based method requires no numerical integration. A similar comparison holds for nonlinear cases too.*

In order to learn a *symplectic* evolution map, we use a tool known as generating functions, which have one-to-one correspondence with symplectic maps. We use a Neural Network, however not for approximating the latent symplectic map, but to approximate its corresponding Generating Function (the method is thus called GFNN). By doing so, the associated evolution map is always symplectic, whether or not it is a good approximation of the latent evolution map, and an appropriate neural network, even just a feedforward one, will be a good approximation after training (see Rmk.3). This symplectic map representation is intrinsic, purely due to the symplectic structure, and no regularization is used.

Moreover, the guaranteed symplecticity originated from the generating function technique allows us to obtain a nontrivial, linearly growing bound on the prediction error:

Theorem 1 (Informal version of Thm.4)**.** *Consider latent dynamics far from chaos (more precisely, being integrable). If the latent generating function is approximated with $\leq \varepsilon$ error in first derivatives, then except for a set of initial conditions whose measure goes to 0 as $\varepsilon \downarrow 0$, the deviation between the predicted sequence $(\boldsymbol{p}_0, \boldsymbol{q}_0), (\boldsymbol{p}_1, \boldsymbol{q}_1,), \ldots$ and the true sequence satisfies*

$$\begin{cases} \|\boldsymbol{p}_n - \boldsymbol{p}(nh)\|_2 \leq C \cdot (nh) \cdot \varepsilon, \\ \|\boldsymbol{q}_n - \boldsymbol{q}(nh)\|_2 \leq C \cdot (nh) \cdot \varepsilon, \end{cases} \quad \forall n \leq h^{-1}\varepsilon^{-1},$$

for some constant $C > 0$, where n is the number of prediction steps and h is the sampling time step of the data.

The merit of this bound lies in long time predictions: note n can be arbitrarily large as ε can be infinitesimal (h is fixed by the training data, and nh is the physical prediction time).

A brief summary of main contributions:

• (Algorithm) *Learn map instead of vector field. Exact symplecticity guaranteed by generating function representation.*

• (Theory) *Linear bound on long-time prediction error.*

• (Validation) *Systematic empirical investigations.*

1.1. Related Works and Discussion

Learning and then predicting dynamics from data is an extremely active research direction. While it is impossible to review all important works, we first mention the classical area of time series (e.g., Box et al., 2015; Abarbanel, 2012; Kantz & Schreiber, 2004; Bradley & Kantz, 2015), where latent differential equations may or may not be assumed. For cases where a latent differential equation is believed to exist, which may correspond to a complex and/or un-modeled underlying dynamical process, some seminal works include (Baake et al., 1992; Bongard & Lipson, 2007; Schmidt & Lipson, 2009), and more recent progress include those based on learning (part of) the vector field via sparse regression of a library (e.g., Brunton et al., 2016; Tran & Ward, 2017; Schaeffer et al., 2018; Lu et al., 2019; Rudy et al., 2017; Kang et al., 2021; Reinbold et al., 2021), learning the vector

[1] Vector-field-based methods can also be designed to make symplectic predictions; see Sec.1.1 4th paragraph.

field via neural network (e.g., Raissi et al., 2018; Rudy et al., 2019; Qin et al., 2019; Long et al., 2018), and learning the vector field via other approaches such as Gaussian processes (e.g., Raissi & Karniadakis, 2018).

'Model-free' approaches that are based on machine learning techniques for sequences have also been proposed, such as (Bailer-Jones et al., 1998) (vanilla RNN), (Wang, 2017) (LSTM), (Pathak et al., 2018) (reservoir computing), (Mukhopadhyay & Banerjee, 2020) (CNN), and (Shalova & Oseledets, 2020) (transformer).

Faced with the extreme success of these generic methods, interests have also been growing in incorporating domain knowledge and specific structures of the underlying problems into the otherwise-black-box schemes (see e.g., Raissi et al., 2019). In terms of mechanical problems modeled by Hamiltonian systems, seminal progress include HNN (Greydanus et al., 2019) and an independent work (Bertalan et al., 2019), SRNN (Chen et al., 2020), SympNets (Jin et al., 2020), and (Lutter et al., 2019; Toth et al., 2020; Zhong et al., 2020; Wu et al., 2020; Xiong et al., 2021), all of which, except SympNets, are related to learning some quantity that produces the Hamiltonian vector field.

In particular, both HNN and SRNN are based on the great idea of learning (using a neural network) the Hamiltonian that generates the vector field (VF), instead of learning the VF itself; this improves accuracy as the Hamiltonian structure of the VF will not be lost due to approximation. HNN learns the Hamiltonian by matching its induced VF with the latent VF (when such information is unavailable, for example in a purely data driven context, data-based approximation such as finite-difference is needed). Then it predicts by numerically integrating the learned VF, and for this we note a Hamiltonian VF doesn't guarantee the symplecticity of its integration[2]. SRNN, on the other hand, learns the Hamiltonian by matching its symplectic integration with the training sequences, and its prediction is then given by symplectic integration of the learned Hamiltonian. It is therefore the closer to GFNN as it essentially learns a symplectic map; it is just that SRNN represents this map by a symplectic integration of a neural-network-approximated Hamiltonian, whereas GFNN represents it by a neural-network-approximated generating function. Because of this, SRNN doesn't need finite-difference approximation and has good prediction accuracy, but it only works for symplectic maps originated from Hamiltonian ODEs, and its accuracy is hampered if the latent Hamiltonian is nonseparable[3].

In comparison, GFNN is not based on Hamiltonian vector fields. It is purely data driven, always symplectic, and works the same for separable-, nonseparable-, or even non-Hamiltonian latent systems.

Worth mentioning is the clever recent work of SympNets (Jin et al., 2020), which also enjoys most of the aforementioned qualitative features of GFNN. It complements GFNN and echoes with our view that directly approximating symplectic maps (instead of Hamiltonian vector fields) in an exactly symplectic way is advantageous (note SRNN can also be seen as a (different) way of doing so). *Algorithmically*, SympNets stack up triangular maps (inspired by symplectic integrator) to construct specialized (new) neural networks, which represent only symplectic maps, and then use them to directly approximate the latent evolution map; GFNN on the other hand uses generating function to indirectly represent the evolution map, and because of its mathematical structure, exact symplecticity is automatically guaranteed, and no special neural network is needed for representing the generating function. Consequently, the *theory* of SympNets is devoted to a universal approximation theorem that characterizes the local prediction error, whereas we focus on the global prediction error (i.e., error after many steps of prediction, instead of one) and rigorously show a nontrivial fact that local errors only accumulate linearly into global error; no approximation theorem needed as it's already established for generic networks. In terms of *performance*, we observe SympNets outperforming vector-field-based approaches (as reasoned above), but GFNN has further improved performance; see e.g., Fig. 2, for which we tried up to 30 layers with 10 sublayers using SympNets' code (both LA- and G-SympNets) and plotted its best result, namely LA-SympNets with 30 layers and 10 sublayers (c.f., here GFNN used 5 layers). We feel SympNets generally require a significantly deeper network than GFNN to achieve high approximation power, but then training and computational challenges may arise.

One more remark is, a good amount of existing work considered predicting chaotic dynamics, but a major part of this work is concerned with structured, non-chaotic dynamics, for which controlled long time (strong) accuracy in individual trajectories becomes possible. Predicting chaos is nontrivial, but sometimes the existence of a chaotic attractor makes the system forgiving, and because of that, prediction errors do not accumulate as much as they can in non-chaotic systems. Besides, one often cares more about statistical accuracy for chaotic systems (e.g., Tsai et al., 2020), as opposed to strong accuracy in trajectory (which usually grows

[2] Unless a symplectic integrator is used. Note the seminal work of HNN used RK45 which is not symplectic, however with small error tolerance (thus good precision but high computation cost).

[3] The original SRNN is based on symplectic integrators for separable Hamiltonians, and nonsymplectic integrators for nonseparable ones; see Footnote 6 for additional information.

too fast in chaos; see Rmk.4). Meanwhile, accurately predicting the trajectory of non-chaotic systems is desirable in numerous applications. Nevertheless, GFNN's predictive power for chaos will be empirically confirmed too.

2. Methods

2.1. Symplectic Map and Generating Function

As we do not assume or seek a latent ODE system but directly approximate the evolution map, a representation of this symplectic map is essential. Instead of directly approximating it, which has the extra difficulty of losing symplecticity (which has to be exact), we use a mathematical tool known as generating function. Let us be more specific:

Firstly, given a (type-2) generating function, there is an associated symplectic map (a.k.a. canonical transformation):

Lemma 1. *Consider a differentiable function $F(\boldsymbol{q}, \boldsymbol{P})$ which shall be called a generating function. The map $[\boldsymbol{p}, \boldsymbol{q}] \mapsto [\boldsymbol{P}, \boldsymbol{Q}]$ implicitly defined by $\boldsymbol{p} = \frac{\partial F}{\partial \boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{P})$, $\boldsymbol{Q} = \frac{\partial F}{\partial \boldsymbol{P}}(\boldsymbol{q}, \boldsymbol{P})$, is a symplectic map.*

Proof. See e.g., (Goldstein, 1980). □

The converse is also true, as long as the latent map doesn't correspond to an evolution time too long (otherwise singularities can be developed):

Lemma 2. *For any infinitesimal symplectomorphism (i.e., symplectic map) on $T^*\mathbb{R}^d$ (i.e., vector phase space), there is a corresponding generating function.*

Proof. This is because the first cohomology group of $T^*\mathbb{R}^d$ is trivial; see e.g., (Da Silva, 2008). □

Remark 1 (generating functions and Hamiltonian system)**.** *We do not assume the latent map that generates the data in discrete time corresponds to an underlying Hamiltonian ODE system in continuous time. There are symplectic maps that do not have such correspondence (see e.g., Sec.4.4).*

On the other hand, given a Hamiltonian system, its flow map, defined as $\phi^t : [\boldsymbol{p}(0), \boldsymbol{q}(0)] \mapsto [\boldsymbol{p}(t), \boldsymbol{q}(t)]$, is symplectic for any t. Therefore, there is a family of corresponding generating functions $F(\boldsymbol{q}, \boldsymbol{P}, t)$, each of which generates the symplectic map $[\boldsymbol{P}, \boldsymbol{Q}] = \phi^t[\boldsymbol{p}, \boldsymbol{q}]$. Moreover, the relation between the Hamiltonian H and F can be made more direct via the Hamilton-Jacobi PDE: $H\left(\frac{\partial F}{\partial \boldsymbol{q}}, \boldsymbol{q}, t\right) + \frac{\partial F}{\partial t} = 0$.

Because of their 1-to-1 correspondence, instead of approximating the symplectic evolution map $\phi : [\boldsymbol{p}, \boldsymbol{q}] \mapsto [\boldsymbol{P}, \boldsymbol{Q}]$, we use a Feedforward Neural Network to approximate the corresponding generating function $F(\boldsymbol{q}, \boldsymbol{P})$. This way, no matter how much error the FNN has in approximating F, it always gives to an evolution map that is *exactly* symplectic.

2.2. Learning Based on Generating Function Training

The type-2 generating function corresponding to a h-time flow map is $F(\boldsymbol{q}, \boldsymbol{P}) = \boldsymbol{q} \cdot \boldsymbol{P} + \mathcal{O}(h)$, and what varies across different problems is inside the $\mathcal{O}(h)$ term. Therefore, for easier training we learn an equivalent, modified generating function S_h, defined through $F(\boldsymbol{q}, \boldsymbol{P}) = \boldsymbol{q} \cdot \boldsymbol{P} + h \cdot S_h(\boldsymbol{q}, \boldsymbol{P})$. It generates a sequence via iteration

$$\begin{cases} \boldsymbol{p}_i = \boldsymbol{p}_{i+1} + h \cdot \partial_1 S_h(\boldsymbol{q}_i, \boldsymbol{p}_{i+1}), \\ \boldsymbol{q}_{i+1} = \boldsymbol{q}_i + h \cdot \partial_2 S_h(\boldsymbol{q}_i, \boldsymbol{p}_{i+1}), \end{cases} \tag{2}$$

as long as an initial condition $[\boldsymbol{p}_0, \boldsymbol{q}_0]$ is provided.

To learn the latent S_h, GFNN uses a neural-network approximation S_h^θ, and trains for a good parameterization θ to best satisfy (2). See Algorithm. 1.

Algorithm 1 GFNN

Data: The data set $\left\{\left([\boldsymbol{p}_{i,j}, \boldsymbol{q}_{i,j}]_{i=0}^{N_j}\right)_{j=1}^{M}\right\}$is observed from sequences generated by a symplectic map ϕ^h, with $[\boldsymbol{p}_{i,j}, \boldsymbol{q}_{i,j}] \in \mathcal{D} \subseteq \mathbb{R}^d \times \mathbb{R}^d \cong T^*\mathbb{R}^d$ and $\boldsymbol{p}_{i+1,j}, \boldsymbol{q}_{i+1,j} = \phi^h(\boldsymbol{p}_{i,j}, \boldsymbol{q}_{i,j})$.
Training: Optimize the loss function

$$\mathcal{L}_{GFNN} = \frac{1}{\sum_{j=1}^{M} N_j} \sum_{j=1}^{M} \sum_{i=0}^{N_j-1} \left(\left\| h\partial_2 S_h^\theta(\boldsymbol{q}_{i,j}, \boldsymbol{p}_{i+1,j}) - (\boldsymbol{q}_{i+1,j} - \boldsymbol{q}_{i,j}) \right\|_2^2 + \left\| h\partial_1 S_h^\theta(\boldsymbol{q}_{i,j}, \boldsymbol{p}_{i+1,j}) - (\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i+1,j}) \right\|_2^2 \right). \tag{3}$$

with respect to neural network parameters θ (see Appendix for our experimental details).
Prediction: Given any initial condition $(\boldsymbol{q}_0, \boldsymbol{p}_0) \in \mathcal{D}$, one step evolution to $(\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{p}}_1)$ can be solved from

$$\begin{cases} \boldsymbol{p}_0 = \tilde{\boldsymbol{p}}_1 + h \cdot \partial_1 S_h^\theta(\boldsymbol{q}_0, \tilde{\boldsymbol{p}}_1), \\ \tilde{\boldsymbol{q}}_1 = \boldsymbol{q}_0 + h \cdot \partial_2 S_h^\theta(\boldsymbol{q}_0, \tilde{\boldsymbol{p}}_1). \end{cases} \tag{4}$$

This can be iterated.

3. Global Error Analysis

We now show that, under reasonable assumptions, GFNN's prediction will be close to the true sequence (continued by the latent ϕ) for a very long time, as a linearly growing long time error bound will be established. This will be contrasted with an obtainable exponentially growing error bound for generic vector-field-based methods. The latter methods are of course more versatile but they do not utilize the special symplectic structure. Proofs are based on normal form and KAM-type techniques and deferred to Appendix.

The main condition needed for this mild error growth is integrability, which, very roughly speaking, requires the latent system to be far from chaos, but it could still be highly nonlinear; see e.g., (Arnol'd, 2013). In order to make it precise, some mathematical preparations are needed, but one can jump to Thm.4 for the main results if preferred.

Definition 1. *A function $g(\boldsymbol{p}, \boldsymbol{q})$ is called a 1st-integral or a constant of motion of the dynamics if it remains constant as $\boldsymbol{p}, \boldsymbol{q}$ evolves in (continuous or discrete) time.*

Definition 2. *The (canonical) Poisson bracket of two arbitrary functions $f(\boldsymbol{p}, \boldsymbol{q}), g(\boldsymbol{p}, \boldsymbol{q})$ is another function defined as $\{f, g\} := \langle \partial f/\partial q, \partial g/\partial p\rangle - \langle \partial f/\partial p, \partial g/\partial q\rangle$.*

Theorem 2 (Arnold-Liouville)**.** *Consider a d-degree-of-freedom Hamiltonian system. Assume there exist d independent 1st integrals in the sense that the Poisson bracket of every pair is 0. If the d-dim. surfaces implicitly defined by the level sets of those 1st integrals are compact, then there exists a canonical transformation from $\boldsymbol{p}, \boldsymbol{q}$ to $\boldsymbol{I}, \boldsymbol{\varphi}$, such that $\boldsymbol{\varphi}$ can be defined on the d-torus, and in the new variables the Hamiltonian only depends on $\boldsymbol{I}$. In this case, $\boldsymbol{I}$, $\boldsymbol{\varphi}$, and the Hamiltonian are respectively called the action, angle variables, and an integrable Hamiltonian.*

Proof. See e.g., (Arnol'd, 2013). □

Remark 2. *In an integrable system, the action variables $\boldsymbol{I}$ remain constants (they are canonical versions of the first integrals), while the angle variables $\boldsymbol{\varphi}$ evolve on an invariant torus $\left\{\boldsymbol{I} = \boldsymbol{I}(0), \boldsymbol{\varphi} \in \mathbb{T}^d\right\}$, where*

$$\mathbb{T}^d = \mathbb{R}^d/(2\pi\mathbb{Z}^d) = \{(\varphi_1, \ldots, \varphi_d) \mod 2\pi; \varphi_i \in \mathbb{R}\}.$$

It is easy to show that the fixed time step generating function for an integrable system takes the form of $S_h(\boldsymbol{I}_0, \boldsymbol{\varphi}_1) = H(\boldsymbol{I}_0)$, as the exact time-$h$ flow is defined as the following

$$\begin{cases} \boldsymbol{I}_1 = \boldsymbol{I}_0, \\ \boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_0 + h\,\partial_1 S_h(\boldsymbol{I}_0, \boldsymbol{\varphi}_1) = \boldsymbol{\varphi}_0 + h\,\nabla H(\boldsymbol{I}_0). \end{cases} \tag{5}$$

Denote $\nabla H(\boldsymbol{I})$ by $\boldsymbol{\omega}(\boldsymbol{I}) = [\omega_1(\boldsymbol{I}), \ldots, \omega_d(\boldsymbol{I})]$. It can be directly seen from Eq. (5) that $\omega_i(\boldsymbol{I})$ represents the change rate (i.e., frequency) of the angle variable φ_i.

Our theory works for almost all initial frequencies, and to describe what are the exceptions we need the following definition, which generalizes irrational numbers in some sense.

Definition 3 ((γ, ν)-Diophantine condition[4])**.** *Frequency vector $\boldsymbol{\omega} = \{\omega_1, \omega_2, \ldots, \omega_d\}$ satisfies (γ, ν)-Diophantine condition iff $|\boldsymbol{k} \cdot \boldsymbol{\omega}| \geq \gamma \cdot \|\boldsymbol{k}\|_1^{-\nu}$, $\forall \boldsymbol{k} \in \mathbb{Z}^d$, $\boldsymbol{k} \neq \boldsymbol{0}$, for some $\gamma > 0, \nu > 0$.*

Definition 4 (ε-neighborhood condition)**.** $(\boldsymbol{p}, \boldsymbol{q}) \in \mathbb{R}^d \times \mathbb{R}^d$ *of an integrable system satisfies ε-neighborhood condition if there exists $\boldsymbol{I}^* \in \mathbb{R}^d$, such that $\boldsymbol{\omega}(\boldsymbol{I}^*)$ satisfies the (γ, ν)-Diophantine condition (Def.3), and $\|\boldsymbol{I}(\boldsymbol{p}, \boldsymbol{q}) - \boldsymbol{I}^*\|_2 \leq c \cdot |\log \varepsilon|^{-\nu-1}$ for some ε independent constant c (defined in the Appendix) with $\boldsymbol{I}(\boldsymbol{p}, \boldsymbol{q})$ being the actions of the system.*

With these preparations, we see the action and angle variables $\boldsymbol{I}, \boldsymbol{\varphi}$ form a new coordinate system alternative to $\boldsymbol{p}, \boldsymbol{q}$ (note even if the system is not integrable and/or time is no longer continuous, one is still free to perform any canonical coordinate transformation; it's just doing so may or may not reveal structured dynamics any more). In fact, they give finer estimates of the prediction error:

Theorem 3 (GFNN's long-time prediction error in actions and angles)**.** *Consider an integrable Hamiltonian system written in action-angle variables, whose exact time-h flow map corresponds to generating function $S_h(\cdot, \cdot)$. Predict its trajectory using GFNN with learned generating function $S_h^\theta(\cdot, \cdot)$ in a bounded data domain $\mathcal{D} = \mathcal{D}_1 \times \mathbb{T}^d \subseteq \mathbb{R}^d \times \mathbb{T}^d$. $\exists\, \varepsilon > 0, \rho > 0$, such that if the learned generating function S_h^θ (extended in a complex neighborhood of $\mathcal{D}$) is analytic and satisfies*

$$\sum_{i=1,2} \left\|\partial_i S_h^\theta(\cdot, \cdot) - \partial_i S_h(\cdot, \cdot)\right\|_\infty \leq C_1 \varepsilon,$$

for some ε independent constant C_1, where the L^∞ norm is defined over the ε independent complex neighborhood $\mathcal{B}_\rho(\mathcal{D})$ of $\mathcal{D}$, then, $\forall (\boldsymbol{I}_0, \boldsymbol{\varphi}_0) \in \mathcal{D}$ that satisfies ε-neighborhood condition (Def.4), the predicted sequence $(\boldsymbol{I}_0, \boldsymbol{\varphi}_0), (\boldsymbol{I}_1, \boldsymbol{\varphi}_1,), \ldots$ generated by GFNN satisfies

$$\begin{cases} \|\boldsymbol{I}_n - \boldsymbol{I}(0)\|_2 \leq C \cdot \varepsilon, \\ \|\boldsymbol{\varphi}_n - \boldsymbol{\varphi}(nh)\|_2 \leq C \cdot (nh) \cdot \varepsilon, \end{cases} \quad \forall n \leq h^{-1}\varepsilon^{-1}, \tag{6}$$

for some constant C.

The intuition behind the proof of Thm. 3 (which is in Appendix) is the following: the predicted dynamics $(\boldsymbol{I}_n, \boldsymbol{\varphi}_n)$ and the true dynamics $(\boldsymbol{I}(nh), \boldsymbol{\varphi}(nh))$ deviate because each step of the prediction introduces some error due to inaccurate S_h^θ, but these errors accumulate in a very delicate way; in fact, earlier errors cannot be amplified too much in order for a linear bound to exist. The key reason, as the proof will recover, is that $\boldsymbol{I}_n$'s dynamics is mostly just oscillatory in time. We show this by decomposing the predicted dynamics into a macroscopic part plus microscopic oscillations. The macroscopic part can be proved to correspond to a barely changing action. The microscopic part, on the other hand, does not accumulate.

Specifically, we introduce a carefully-chosen near-identity canonical coordinate change $\mathcal{T} : [\boldsymbol{I}, \boldsymbol{\varphi}] \mapsto [\boldsymbol{J}, \boldsymbol{\theta}]$, $\mathcal{T} \approx id + \mathcal{O}(\varepsilon)$, and show that the new variables $[\boldsymbol{J}_n, \boldsymbol{\theta}_n]$ describe, roughly, the macroscopic part of the predicted dynamics.

[4] also known as strong non-resonance condition

We then prove, when compared to the true dynamics,

$$\begin{cases} \|\boldsymbol{I}(nh) - \boldsymbol{J}_n\|_2 = \mathcal{O}\left(\epsilon\right), \\ \|\boldsymbol{\varphi}(nh) - \boldsymbol{\theta}_n\|_2 = nh \cdot \mathcal{O}(\varepsilon), \end{cases} \quad \forall nh = \mathcal{O}\left(\varepsilon^{-1}\right).$$

Since $\mathcal{T}$ is near-identity, $[\boldsymbol{J}_n, \boldsymbol{\theta}_n] = [\boldsymbol{I}_n, \boldsymbol{\varphi}_n] + \mathcal{O}(\varepsilon)$ for all n, and the triangle inequality then completes the proof. □

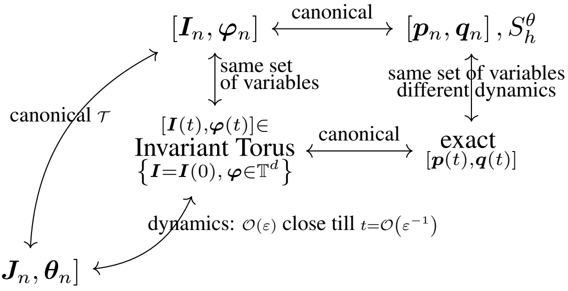
$[\boldsymbol{I}_n, \boldsymbol{\varphi}_n]$
canonical
$[\boldsymbol{p}_n, \boldsymbol{q}_n], S_h^\theta$
same set of variables
same set of variables different dynamics
canonical $\mathcal{T}$
$[\boldsymbol{I}(t),\boldsymbol{\varphi}(t)]\in$ Invariant Torus $\left\{\boldsymbol{I}=\boldsymbol{I}(0), \boldsymbol{\varphi}\in\mathbb{T}^d\right\}$
canonical
exact $[\boldsymbol{p}(t),\boldsymbol{q}(t)]$
dynamics: $\mathcal{O}(\varepsilon)$ close till $t=\mathcal{O}(\varepsilon^{-1})$
$\exists\,[\boldsymbol{J}_n, \boldsymbol{\theta}_n]$

Figure 1. Main components in the proof of linear error growth.

Now we can relate the error bound in Thm. 3 back to that for the original variables $\boldsymbol{p}, \boldsymbol{q}$. The big picture is summarized by Fig.1. In the end, our theory only requires the existence of action and angle variables, and no knowledge about how to find the actions and angles is needed.

Theorem 4 (linear growth of GFNN long-time prediction error)**.** *Consider an integrable Hamiltonian system whose exact solution is denoted by $\boldsymbol{p}(t), \boldsymbol{q}(t) \in \mathbb{R}^d$. Denote by $S_h(\cdot, \cdot)$ the generating function corresponding to its exact time-h flow map. Consider predicting its trajectory using GFNN with learned generating function $S_h^\theta(\cdot, \cdot)$ in a bounded data domain $\mathcal{D} \subseteq \mathbb{R}^d \times \mathbb{R}^d$. $\exists\, \varepsilon > 0, \rho > 0$, such that if the learned generating function S_h^θ (extended in a complex neighborhood of $\mathcal{D}$) is analytic and satisfies*

$$\sum_{i=1,2} \left\|\partial_i S_h^\theta(\cdot, \cdot) - \partial_i S_h(\cdot, \cdot)\right\|_\infty \le C_1 \varepsilon, \qquad (7)$$

for some ε independent constant C_1, where the L^∞ norm is defined over the ε independent complex neighborhood $\mathcal{B}_\rho(\mathcal{D})$ of $\mathcal{D}$, then, $\forall\, (\boldsymbol{p}_0, \boldsymbol{q}_0) \in \mathcal{D}$ that satisfies ε-neighborhood condition (Def.4) with nonlinear frequency $\boldsymbol{\omega}(\cdot)$ being given by S_h, the predicted sequence $(\boldsymbol{p}_0, \boldsymbol{q}_0), (\boldsymbol{p}_1, \boldsymbol{q}_1,), \ldots$ generated by GFNN satisfies

$$\begin{cases} \|\boldsymbol{p}_n - \boldsymbol{p}(nh)\|_2 \le C \cdot (nh) \cdot \varepsilon, \\ \|\boldsymbol{q}_n - \boldsymbol{q}(nh)\|_2 \le C \cdot (nh) \cdot \varepsilon, \end{cases} \quad \forall n \le h^{-1}\varepsilon^{-1}, \quad (8)$$

for some constant $C > 0$.

Remark 3. *It is known that neural networks can approximate functions and their derivatives with any precision; see e.g., the classical work (Hornik et al., 1990) and a more recent discussion (Yarotsky, 2017). (7) can thus be attained.*

Remark 4. *The integrability assumption in Thm.4 is nontrivial, however reasonable. This is because it rules out the possibility of a positive Lyapunov exponent, which by definition indicates that a deviation between two trajectories can exponentially grow in time (e.g., Alligood et al., 1996). Naturally, if the latent system does have a positive Lyapunov exponent, then in general one should not expect a linearly growing prediction error, as an arbitrarily small approximation error, even if it's just made in one step, can be exponentially amplified.*

A simple illustration of this is a Hamiltonian system $\dot{x} = y, \dot{y} = x$, which is not integrable due to noncompactness (not even chaos). It has a Lyapunov exponent of $+1$. Consider predictions based on approximation $\dot{x} = y + \delta_x, \dot{y} = x + \delta_y$, then no matter how small δ_x and δ_y are, the difference between its solution and the original one grows like $\exp(t)$ except for measure zero δ_x and δ_y values.

As a comparison, if the prediction map is not symplectic, either due to nonsymplectic numerical integration, or because the learned vector field is no longer Hamiltonian, local prediction error (in each step) may get amplified and long time prediction error may grow exponentially:

Theorem 5. *Consider the latent dynamics $\dot{x} = f(x)$ and its prediction via an Euler integration of the learned vector field $x_{i+1} = x_i + h\tilde{f}(x_i)$, with consistent initial condition $x(0) = x_0$. Assume f is L-Lipschitz continuous, $\mathcal{C}^1$, the learned vector field is accurate up to δ in the sense that $\|\tilde{f} - f\|_\infty \le \delta$, and the prediction remains bounded. Then the accuracy of the prediction at time $T = nh$ satisfies*

$$\|x(T) - x_n\| \le \frac{\exp(LT) - 1}{L}(\delta + Lh/2).$$

Proof. See Appendix. □

Remark 5. *This exponential growth with T (and n) is not an overestimation. A simple example that attains it is $f(x) = x$ and $\tilde{f}(x) = x + \delta$. This is of course because the latent dynamics is structurally bad and does not forgive past errors, but that is exactly our point: when the latent dynamics has specific structures such as being a symplectic flow, utilizing those structures in the prediction could lead to much better controlled accumulation of errors.*

Remark 6. *In the context of learning dynamics from data, two sources contribute to the difference between f and $\tilde{f}$. One is approximation error, for instance of the neural network; the other is because one doesn't have an oracle about the latent vector field f but only its approximation from the data, for example $f(x_i) \approx (x_{i+1} - x_i)/h$. A map based approach doesn't directly use f and thus can avoid the latter error, and it doesn't have numerical integration errors in the next phase of predictions either. A neural ODE type* *treatment (Zhong et al., 2020) can avoid the latter error too, but integration errors in the prediction phase remain (unless computationally expensive small steps are used).*

4. Experiments

Let's now systematically (within the page limit) investigate the empirical performances of GFNN. It was conjectured that invariant sets of a smooth map with a dense trajectory are typically either periodic, quasiperiodic[5], or chaotic (Sander & Yorke, 2015). Thus, Sec.4.1-4.4 will study classical examples that respectively correspond to periodic, quasiperiodic+chaotic, quasiperiodic, and quasiperiodic+chaotic cases. We'll see smaller and linearly growing errors of GFNN in both periodic and quasiperiodic cases, even when the latent system is not integrable. In chaotic cases, GFNN will also exhibit pleasant statistical accuracy.

VFNN stands for: learning the Vector Field via a Neural Network (without caring about the Hamiltonian structure).

Details of data preparation and training are in appendix B.

4.1. An Integrable and Separable Hamiltonian: 2-Body Problem

Consider the motion of 2 gravitationally interacting bodies. Letting their distance be $\boldsymbol{q}(t)$ and the corresponding momentum be $\boldsymbol{p}(t)$, the problem can be equivalently turned into (after unit normalization) an ODE system governed by

$$H\left(\boldsymbol{p}, \boldsymbol{q}\right) = \left\|\boldsymbol{p}\right\|_2^2/2 - 1/\left\|\boldsymbol{q}\right\|_2 .$$

Despite its high nonlinearity, this is an integrable system. Analytical solutions known as Keplerian orbits exist and are periodic in bounded cases. Each solution is described by important physical quantities known as orbital elements, which include semi-major axis and eccentricity, that characterize the shape of the elliptic orbit. As shown in Fig. 2, GFNN outperforms other methods and keeps the errors of semi-major axis and eccentricity small and bounded, which is consistent with Thm. 3 because semi-major axis and eccentricity are functions of actions known as Delaunay variables (Morbidelli, 2002). The advantage of GFNN can also been seen in the original variables (e.g., $\boldsymbol{q}$), and the zoomed-in plots in row 2 show that the next two top performers are SRNN (seq_len=2) and SRNN (seq_len=5); SympNets has notably larger error in the orbital phase but its accuracy in the orbital shape is actually comparable to SRNN.

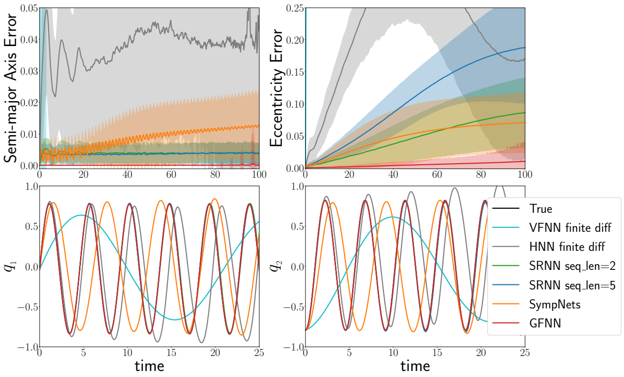

Figure 2. Comparison of 2D Keplerian orbit predicted by different methods. The 1st row is the error growth of two variables of physical importance, namely semi-major axis and eccentricity (for this problem, their true values are both constants). Mean values of prediction errors starting from 1,000 i.i.d. initial conditions are plotted with shades corresponding to 1 standard deviation. The 2nd row zooms in the position variables of one predicted sequence (in q_1 and q_2 respectively). Data sequences are prepared with time step 0.1.

4.2. A Non-integrable but Separable Hamiltonian System: Hénon-Heiles

The Hénon-Heiles system describes the motion of stars around a galactic center (Hénon & Heiles, 1964). It is a classical non-integrable system with very complex dynamics, governed by Hamiltonian

$$H\left(p_1, p_2, q_1, q_2\right) = \frac{\left(p_1^2 + p_2^2\right)}{2} + \left(\frac{q_1^2 + q_2^2}{2} + q_1^2 q_2 + \frac{q_2^3}{3}\right).$$

Both chaotic and (quasi)-periodic solutions exist. Initial conditions corresponding to higher energy (i.e., H's value) are more likely to be chaotic. We investigate GFNN's performance in both cases.

4.2.1. Non-periodic but regular motions

For a non-chaotic initial condition, numerically observed was that the long time prediction error of GFNN still grows linearly even though the latent system is no longer integrable; see Fig. 3. Notably, SRNN also exhibits linear error growth (although at a higher rate), and this is consistent with our intuition as SRNN also learns a symplectic evolution map (indirectly via the symplectic integration of a Hamiltonian to-be-learned). HNN, on the other hand, has exponentially growing error which quickly saturates to maximum values (due to boundedness of trajectories).

GFNN's linear error growth despite non-integrability is due to the existence of mathematical objects known as KAM-tori (e.g., Pöschel, 1982). They correspond to part of the phase space where dynamics are topologically equivalent to integrable ones. A by-product is, solutions in this region are

[5] A function $f(t)$ is quasiperiodic if $\exists$ some constants $n \in \mathbb{Z}^+$, $\Omega_1, \cdots, \Omega_n \in \mathbb{R}$, and some function F 1-periodic in each argument, s.t., $f(t) = F(\Omega_1 t, \cdots, \Omega_n t)$. An integrable system's solution is quasiperiodic if $\mathrm{LCM}(\omega_1(\boldsymbol{I}), \cdots, \omega_d(\boldsymbol{I}))$ doesn't exist (see Rmk.2 for $\omega_i(\boldsymbol{I})$); otherwise it is periodic.

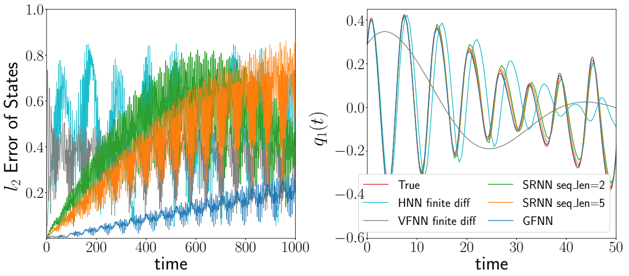

Figure 3. Error of predictions of a quasi-periodic trajectory with energy level near $\frac{1}{12}$ of the Hénon-Heiles system. Data sequences are prepared with time step 0.5.

either quasiperiodic or periodic (see Footnote 5).

4.2.2. Dynamics in chaotic sea

To visualize the prediction of chaotic dynamics, which take place in 4D, we use the standard tool of Poincaré section, which plots where an orbit intersects with a 2D slice of the 3D constant-energy manifold. Fig. 4 shows the Poincaré section produced by predictions of different methods, based on the same initial condition that leads to chaotic motion via the latent dynamics. The true chaotic motion is ergodic on a submanifold of the phase space, and when restricting to the Poincaré section, it gives intersections that are dense in a subset known as the chaotic sea. Therefore, the shapes of the dense area and the holes inside it (often corresponding to regular islands on which motions are (quasi)-periodic) are indicators of the prediction accuracy. Among methods tested in Fig. 4, only VFNN didn't produce a pattern similar to the truth. Quantitative comparisons are conducted by comparing the empirical distributions of points on the Poincaré section, and KL divergences between their marginals and the truth are annotated along with the histograms. GFNN has the smallest errors.

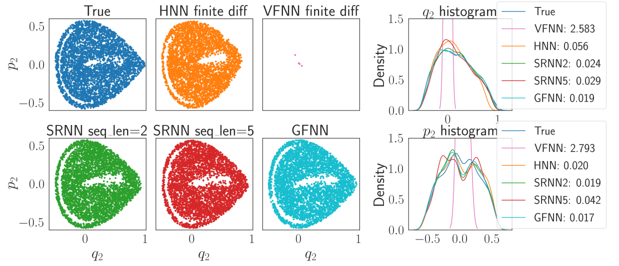

Figure 4. Quantifying the statistical accuracy in predicting a chaotic orbit of the Hénon-Heiles system. Left 3 columns: Poincaré section; right column: marginal distributions and their KL divergences from the truth. The plotted orbit corresponds to energy $\frac{1}{6}$; Poincaré section is given by $q_2(t)$, $p_2(t)$ at $q_1(t) = 0$. Data sequences are prepared with time step 0.5.

4.3. A Non-integrable and Non-separable Hamiltonian: Planar Circular Restricted 3-Body Problem (PCR3BP)

PCR3BP is a special case of the gravitational 3-body problem. In addition to a co-planar restriction, it assumes two bodies massive and the third infinitesimal, which models settings like mission design for a space shuttle near Earth and Moon (Koon et al., 2000), and understanding a planet's motion around binary stars (Li et al., 2016; Quarles et al., 2020). Its Hamiltonian takes the form

$$\begin{aligned} H(\boldsymbol{p},\boldsymbol{q}) = \frac{p_1^2+p_2^2}{2} + p_1 q_2 - p_2 q_1 \\ - \frac{1-\mu}{\|(q_1+\mu, q_2)\|_2} - \frac{\mu}{\|(q_1+\mu-1, q_2)\|_2}, \end{aligned}$$

with $\mu \in (0,1)$ a constant mass parameter. Note it cannot be written as $K(\boldsymbol{p}) + V(\boldsymbol{q})$, hence nonseparable.

In order to focus on comparing with SOTA methods for trajectory accuracy, we predict solutions in the nearly-integrable (non-chaotic) regime of PCR3BP; see Fig.5. GFNN still has the smallest error among those experimented, and its growth is again linear. SRNN typically performs the best among tested existing approaches, but its published version loses symplecticity in this case due to non-separability[6], and its accuracy deteriorated. Note also that for methods that learn, in the separable case, $V(\boldsymbol{q})$ in the Hamiltonian or $\nabla V(\boldsymbol{q})$ in the vector field, now they cannot just do so but have to learn the entire $H(\boldsymbol{p},\boldsymbol{q})$ in doubled dimensions.

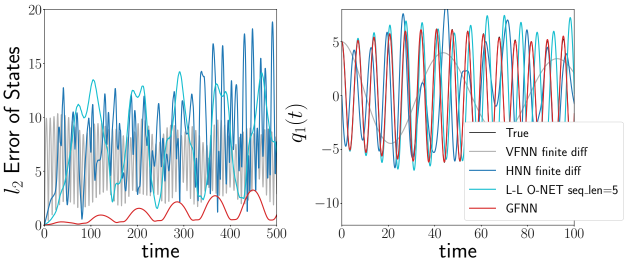

Figure 5. Comparison of PCR3BP orbit predicted by different methods. L-L O-NET (in SRNN paper) is selected instead of SRNN as the Hamiltonian is not separable. Data sequences are prepared with time step 0.1.

4.4. A Discrete-time Non-(Smooth-)Hamiltonian System: the Standard Map

The standard map is a classical model in accelerator physics. It is a chaotic system whose statistical property is (relatively)

[6] A possible remedy based on our nonseparable symplectic integrators (Tao, 2016) was mentioned in SRNN as a future direction. This remedy is implemented in a concurrent work (Xiong et al., 2021), which successfully reduces the error of predicting nonseparable dynamics to the level of SRNN for separable dynamics.

well understood. It is a symplectic map in 2D given by

$$\begin{cases} p_{n+1} &= p_n + K \sin(\theta_n), \\ \theta_{n+1} &= \theta_n + p_{n+1}. \end{cases} \tag{9}$$

The dynamics is genuinely in *discrete* time, as no smooth Hamiltonian ODE can produce a flow map like it[7]. K is a positive constant that controls the strength of nonlinearity, and it has been estimated that the region of initial conditions leading to chaos has size increasing with K (Chirikov, 1979).

Methods based on vector fields (e.g., VFNN) or Hamiltonian (e.g., HNN, SRNN) are not very suitable for this prediction task because there is no latent continuous (Hamiltonian) dynamics. One can still apply these methods regardless, for example by using finite differences to construct a fictitious vector field for VFNN and HNN to learn, or just use SRNN without realizing that no Hamiltonian will be able to produce the training data. Their results (obtained using $h = 1$) will be compared with those of GFNN, which is still applicable here as it directly learns evolution maps.

Fig. 6 illustrates the predicted evolutions of a fixed initial condition in the chaotic sea (of the true dynamics, $K = 1.2$) by various methods. Note both θ and p have been mod 2π as this quotient compactifies the phase space into the 2-torus without affecting the dynamics (see Eq. (9)). Like before, the prediction quality can be inferred from the geometric shape of the set of plotted points, which should match that of the truth (i.e., the latent chaotic sea), and quantitative comparisons can be made using distances between empirical distributions of p, θ values collected along long time predictions (KL divergences from the truth are provided).

One can see GFNN is the only method that captures the major regular islands (the big holes), but even GFNN does not capture the minor regular islands well. The standard map seems to be a challenging problem; HNN and SRNN did not manage to reproduce any chaotic motion, and VFNN completely distorted the chaotic sea.

Fig. 7 on the other hand illustrates predictions in regular islands (of the true dynamics, $K = 0.6$). The two (not three, note periodic boundary conditions) elliptical shapes near $p \approx \pi$ and $\theta \approx 0, \pi$ correspond to quasiperiodic orbits, and GFNN is the only one that captures them: the exact trajectory is jumping back and forth between two islands, so does GFNN's prediction, while other methods tend to produce continuous trajectories without capturing the jumps.

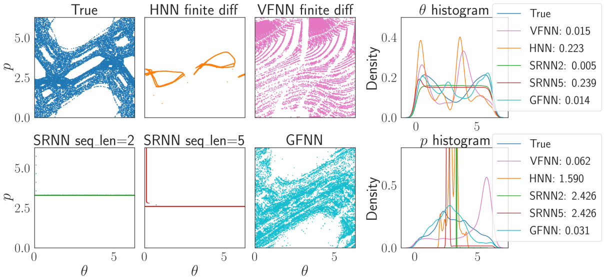

Figure 6. Predict a chaotic orbit of the standard map. Left 3 columns: the predicted orbit in phase space; right column: marginals of its empirical measure and their KL divergences from the truth.

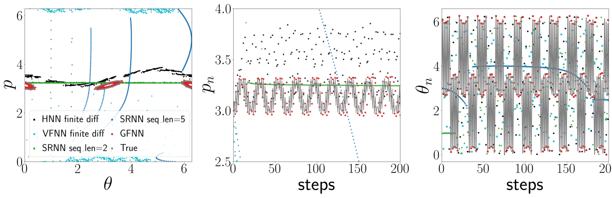

Figure 7. Predict a regular orbit of the standard map. The 1st plot is orbits predicted by various methods in phase space. The 2nd and 3rd plots shows how their two coordinates change with time.

Acknowledgements

The authors thank the anonymous reviewers who helped significantly improve the quality of this article, and Minshuo Chen for helpful discussions. We are grateful for the partial support by NSF DMS-1847802 and ECCS-1936776.

References

Abarbanel, H. *Analysis of observed chaotic data*. Springer Science & Business Media, 2012.

Alligood, K. T., Sauer, T. D., and Yorke, J. A. *Chaos*. Springer, 1996.

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

Baake, E., Baake, M., Bock, H., and Briggs, K. Fitting ordinary differential equations to chaotic data. *Physical Review A*, 45(8):5524, 1992.

Bailer-Jones, C. A., MacKay, D. J., and Withers, P. J. A recurrent neural network for modelling dynamical systems. *network: computation in neural systems*, 9(4):531–547, 1998.

Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G. On learning hamiltonian systems from data. *Chaos: An*

[7] This is because autonomous Hamiltonian systems in 2D are never chaotic (the Hamiltonian itself is a 1st integral) but the standard map is chaotic.

Interdisciplinary Journal of Nonlinear Science, 29(12): 121107, 2019.

Bongard, J. and Lipson, H. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Bradley, E. and Kantz, H. Nonlinear time-series analysis revisited. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097610, 2015.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020.

Chirikov, B. V. A universal instability of many-dimensional oscillator systems. *Physics reports*, 52(5):263–379, 1979.

Da Silva, A. C. *Lectures on symplectic geometry*, volume 3575. Springer, 2008.

Goldstein, H. Classical mechanics, 1980.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. *NeurIPS*, 2019.

Hairer, E., Lubich, C., and Wanner, G. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

Hénon, M. and Heiles, C. The applicability of the third integral of motion: some numerical experiments. *The Astronomical Journal*, 69:73, 1964.

Hornik, K., Stinchcombe, M., and White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

Kang, S. H., Liao, W., and Liu, Y. IDENT: Identifying differential equations with numerical time evolution. *Journal of Scientific Computing*, 2021.

Kantz, H. and Schreiber, T. *Nonlinear time series analysis*, volume 7. Cambridge university press, 2004.

Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D. Heteroclinic connections between periodic orbits and resonance transitions in celestial mechanics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 10(2): 427–469, 2000.

Li, G., Holman, M. J., and Tao, M. Uncovering circumbinary planetary architectural properties from selection biases. *The Astrophysical Journal*, 831(1):96, 2016.

Long, Z., Lu, Y., Ma, X., and Dong, B. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pp. 3208–3216. PMLR, 2018.

Lu, F., Zhong, M., Tang, S., and Maggioni, M. Nonparametric inference of interaction laws in systems of agents from trajectory data. *Proceedings of the National Academy of Sciences*, 116(29):14424–14433, 2019.

Lutter, M., Ritter, C., and Peters, J. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019.

Morbidelli, A. *Modern celestial mechanics: aspects of solar system dynamics*. 2002.

Mukhopadhyay, S. and Banerjee, S. Learning dynamical systems in noise using convolutional neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(10):103125, 2020.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

Pöschel, J. Integrability of Hamiltonian systems on cantor sets. *Communications on Pure and Applied Mathematics*, 35(5):653–696, 1982.

Qin, T., Wu, K., and Xiu, D. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.

Quarles, B., Li, G., Kostov, V., and Haghighipour, N. Orbital stability of circumstellar planets in binary systems. *The Astronomical Journal*, 159(3):80, 2020.

Raissi, M. and Karniadakis, G. E. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Reinbold, P. A., Kageorge, L. M., Schatz, M. F., and Grigoriev, R. O. Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nature Communications*, 12(1):1–8, 2021.

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

Rudy, S. H., Kutz, J. N., and Brunton, S. L. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *Journal of Computational Physics*, 396:483–506, 2019.

Sander, E. and Yorke, J. A. The many facets of chaos. *International Journal of Bifurcation and Chaos*, 25(04): 1530011, 2015.

Schaeffer, H., Tran, G., and Ward, R. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295, 2018.

Schmidt, M. and Lipson, H. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

Shalova, A. and Oseledets, I. Tensorized transformer for dynamical systems modeling. *arXiv preprint arXiv:2006.03445*, 2020.

Stein, E. M. and Shakarchi, R. *Complex analysis*, volume 2. Princeton University Press, 2010.

Tao, M. Explicit symplectic approximation of nonseparable hamiltonians: Algorithm and long time performance. *Physical Review E*, 94(4):043303, 2016.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020.

Tran, G. and Ward, R. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3):1108–1129, 2017.

Tsai, S.-T., Kuo, E.-J., and Tiwary, P. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature Communications*, 11(1):1–11, 2020.

Wang, Y. A new concept using lstm neural networks for dynamic system identification. In *2017 American Control Conference (ACC)*, pp. 5324–5329. IEEE, 2017.

Wu, K., Qin, T., and Xiu, D. Structure-preserving method for reconstructing unknown hamiltonian systems from trajectory data. *SIAM Journal on Scientific Computing*, 42(6):A3704–A3729, 2020.

Xiong, S., Tong, Y., He, X., Yang, S., Yang, C., and Zhu, B. Nonseparable symplectic neural networks. *ICLR*, 2021.

Yarotsky, D. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.