Leap-frog neural network for learning the symplectic evolution from partitioned data

Xin Li\textsuperscript{1}, Jian Li\textsuperscript{2,3} and Zhihong Jeff Xia\textsuperscript{4}

\textsuperscript{1} Department of Statistics and Data Science, Southern University of Science and Technology of China. No 1088, Xueyuan Rd., Xili, Nanshan District, Shenzhen, Guangdong, 518055, PR China, 
\textsuperscript{2} School of Astronomy and Space Science, Nanjing University, 163 Xianlin Avenue, Nanjing 210023, PR China, 
\textsuperscript{3} Key Laboratory of Modern Astronomy and Astrophysics in Ministry of Education, Nanjing University, Nanjing 210023, PR China 
\textsuperscript{4} Department of Mathematics, Northwestern University, 2033 Sheridan Road, Evanston, IL 60208, USA

Accepted 1988 December 15. Received 1988 December 14; in original form 1988 October 11

ABSTRACT

For the Hamiltonian system, this work considers the learning and prediction of the position ($q$) and momentum ($p$) variables generated by a symplectic evolution map. Similar to Chen & Tao (2021), the symplectic map is represented by the generating function. In addition, we develop a new learning scheme by splitting the time series ($q_i, p_i$) into several partitions, and then train a leap-frog neural network (LFNN) to approximate the generating function between the first (i.e. initial condition) and one of the rest partitions. For predicting the system evolution in a short timescale, the LFNN could effectively avoid the issue of accumulative error. Then the LFNN is applied to learn the behavior of the 2:3 resonant Kuiper belt objects, in a much longer time period, and there are two significant improvements on the neural network constructed in our previous work (Li et al. 2022): (1) conservation of the Jacobi integral; (2) highly accurate prediction of the orbital evolution. We propose that the LFNN may be useful to make the prediction of the long time evolution of the Hamiltonian system.

Key words: celestial mechanics – Kuiper belt: general – planets and satellites: dynamical evolution and stability – methods: miscellaneous

1 INTRODUCTION

Neural networks were designed for pattern recognition by learning the proper function between covariates (i.e. inputs) and variates (i.e. outputs) as a black-box (McCulloch & Pitts 1943; Rosenblatt 1958). The aim of artificial neural networks (ANNs) is to solve complicated problems by the method inspired from human brain. The ANN, which consists of an interconnected network, can learn from a large amount of experience data by updating the weights on its connections. Such a strategy has been efficiently applied to a wide range of pattern recognition problems in science and industry.

Time series prediction has become an intensive area of academic research. Traditional methods have constructed parametric models informed by domain expertise, e.g. autoregressive (AR) (Box & Jenkins 1976), exponential smoothing (Winters 1960; Gardner 1985) and structural time series models (Harvey 1990). After that, machine learning methods provide a way to learn temporal dynamics in a data-driven manner (Ahmed et al. 2010). The significant increase in computing power has an enormous effect on the area of the time series forecasting, as machine learning now plays a key role in this regard.

Traditionally, differential equations are widely used for most physical models. In order to predict the time series, the modern machine learning—or, more precisely, the deep ANN (i.e., with more than one hidden layer)—is trained to represent continuous dynamical systems (E 2017). One of the major tasks in deep learning is to learn the solutions of the differential equations as time series; and to go a step further, another advanced task it to learn the differential equations themselves, representing by a class of functions.

In the field of astronomy and astrophysics, the method of machine learning has already been widely applied to exoplanet discovery (Schanche et al. 2019; Armstrong, Gamper & Damoulas 2021), galaxy classification (Zhang et al. 2019; Baqui et al. 2021; Vavilova et al. 2021), research of dark matter (Agarwal, Dave & Bassett 2018; Lucie-Smith, Peiris & Pontzen 2019; Petulante et al. 2021). However, for the classical n-body problem in celestial mechanics, the relative study did not start until very recently.
Dzamba & Yosinski (2019) studied integrable Hamiltonian systems where the motion is regular everywhere, e.g. the two-body problem. They designed “Hamiltonian neural networks (HNN)” to learn the laws of physics directly from data, and also the conservation of the energy-like quantities on a long timescale. They applied the trained HNN to predict the position and momentum variables \((q,p)\) of future orbits. Nevertheless, their method doesn’t work very well on the 3-body problem, which is actually non-integrable. A few months later, Breen et al. (2020) declared that they could solve the general 3-body problem by means of training a deep ANN. For the planar case, they demonstrated that the deep ANN can provide solutions as accurate as those from the numerical integration within much less computational time. While we notice that their best-performing ANN is trained for trajectories with a relatively short evolution time of \(T = 3.9\) time units.

The fundamental difficulty in learning the motion of the 3-body problem is that, the system could be chaotic, i.e. the slight changes in the initial conditions can lead to significantly different long-term trajectories (Poincaré 1892). In the framework of the planar restricted 3-body problem, we studied the behaviors of objects in the 2:3 mean motion resonance with Neptune, which can be considered as an intermediate case of regular motion (Li et al. 2022). By learning the solutions of the differential equations, in terms of the orbital elements, our designed ANN can predict the trajectories of the 2:3 resonators as long as a entire resonant period, over \(\sim 20000\) yr. Initially, we tried to use the position and momentum variables as in previous works, but we found that the predicted solutions can not guarantee the conservation of the Jacobi integral. Then we turned to use the orbital elements as training data, because in this way we actually assumed that the system is nearly the perturbed two-body problem. With this trick, we could predict the long-term trajectory along which the Jacobi integral is almost conserved. However, the weakness of employing the orbital elements is that, the semimajor axis and eccentricity are always on the order of 0.1-1, while the other two angles could increase to the values \(\gg 1\) when the evolution time is long, then the difference in the orders of magnitudes of these quantities would not be suitable for the training process. So we were wondering, if our training data set comprising position and momentum variables, which are always on the same order of magnitude, how to make sure the Jacobi integral is conserved for the predicted orbits, i.e. keeping the system symplectic.

Very recently, Chen & Tao (2021) considered the learning and prediction of the non-linear time series generated by a latent symplectic map. They studied the Hamiltonian systems whose solution flows give such symplectic maps. They predicted the future trajectory by representing the symplectic map via a generating function, which is approximated by the so-called generating function neural network (GFNN). Chen & Tao (2021) has compared their method with some other works which perform quite well in learning the Hamiltonian systems, e.g. HNN (Greydanus, Dzamba & Yosinski 2019) and an independent work (Bertalan et al. 2019), SRNN (Chen et al. 2020), SympNets (Jin et al. 2020), and Lutter, Ritter & Peters (2019), Toth et al. (2020), Zhong, Dev & Chakraborty (2020), Wu, Qin & Xiu (2020), Xiong et al. (2021), etc. All of these networks, except SympNets, are designed to learn some quantities that produce the Hamiltonian vector field. While the GFNN applies generating function along the evolution step by step, and the results show that it performs better on several examples compared to the other works mentioned above; while an intrinsic issue is that, the accumulation of error grows linearly in comparison with the ground truth obtained by the numerical simulation.

We then are thinking about that whether we could develop the idea “generating function” for learning the symplectic evolution from the position and momentum variables. We are about to make a major adjustment to Chen & Tao (2021): for the trajectories in the training set, we divide each trajectory into several partitions, and then deduce the generating function between the first and one of the rest partition elements. In this way, the machine could learn a class of generating functions for the entire trajectory, i.e. the time series evolution. Since each of the partition elements, at different time windows, is equally (symplectically) mapped from the first one, the accuracy of our prediction should be kept in a similar level but not increase with time. As we will show later, for the same data set of the 3-body problem provided by Chen & Tao (2021), the overall loss of our method can achieve a value that 2-3 times smaller than that of the GFNN. At this point, we may solve the problem raised in our previous work (Li et al. 2022), that we can use the training data consisting of the positions \(q\) and the momenta \(p\) to predict the long future orbits with the required accuracy, and also make sure the evolution is symplectic, i.e. keeping the Jacobi integral constant.

The rest of this paper is organised as follows: in Section 2, we introduce the mathematical background about linear Hamiltonian systems, including the symplectic map and generating function, and how to design the ANN to predict the future time series. In Section 3, for a dynamical model in the framework of planar restricted 3-body problem, we compare the performance of our designed ANN with the GFNN from Chen & Tao (2021). In Section 4, we apply this ANN to the evolution of the non-resonant and resonant Kuiper belt objects (KBOs), and show that the orbit prediction does have stable error even for quite a long time interval, and the Jacobi integral is also well conserved. The conclusions and discussion are presented in Section 5.

2 METHODS

2.1 Hamiltonian system

We first review some basic mathematical background of Hamiltonian systems (Meyer, Hall & Offin 2009). Let \(gl(m, \mathbb{R})\) denote the set of all \(m \times m\) non-singular matrices with entries in the real number field \(\mathbb{R}\). Let \(z\) be a coordinate vector in \(\mathbb{R}^2n\), \(I\) be a time interval in \(\mathbb{R}\), and the matrix \(K: I \to gl(2n, \mathbb{R})\) be continuous and symmetric. Then a Hamiltonian system can be described by \(2n\) ordinary differential equations:

\[
\dot{z} = J \frac{\partial H}{\partial z} = JK(t)z = A(t)z, \tag{1}
\]

where \(H = H(z, t)\) is called the Hamiltonian of the system, and \(J\) is the \(2n \times 2n\) Poisson matrix:

\[
J = \begin{bmatrix} 0 & -I_n \\ I_n & 0 \end{bmatrix}, \tag{2}
\]
where $I_n$ is the $n \times n$ identity matrix.

Let $t_0 \in I$ be the initial time point. From the theory of differential equations, if the right function $A(t)z$ is $C^1$, or the Lipschitz condition in the variable $z$ is satisfied, given the initial condition $z_0 = z(t_0) \in \mathbb{R}^n$, there exists a unique solution $z(t, t_0, z_0)$ of equation (1) for $t \in I$. We say the time evolution of $z$ is symplectic if and only if the Jacobi matrix of the transformation $z'$ satisfies

$$(z')^T J (z') = J. \tag{3}$$

As for the Hamilton’s equation (1), this condition is naturally satisfied.

In the current work, the latent evolution of a Hamiltonian system is considered. Let $z(t) = (q(t), p(t))$, where $q$ is called the coordinate and $p$ is called momentum, and these two variables are said to be conjugated. Then the system of equation (1) can be written in the form

$$\dot{p}(t) = -\frac{\partial H}{\partial q(t)}, \quad \dot{q}(t) = \frac{\partial H}{\partial p(t)}. \tag{5}$$

Assuming each time series $(q(t), p(t))$ is sampled at discretized time points, with a time step $h > 0$, then we have

$$q_i = q(t_i), \quad p_i = p(t_i) \quad (t_i = ih). \tag{6}$$

Now we consider the one step mapping

$$\phi: (q(t_i), p(t_i)) \rightarrow (q(t_{i+1}), p(t_{i+1})), \quad \forall t_i. \tag{7}$$

Since the time series $(q(t), p(t))$ is symplectic, the corresponding mapping $\phi$ is symplectic.

In our previous work [Li et al. 2022], the solution of the restricted 3-body problem is learned by multi-layer neural networks. We tried to train the ANN using the data of positions and velocities (equivalent to $q$ and $p$, respectively). Although the prediction could be very closed to the ground truth data, the Jacobi integral seems to vary greatly along a small scale. Indeed, we can also say $\phi$ in equation (7) is a canonical transformation, then we will adopt the classical technique “generating function” in the following.

### 2.2 Symplectic mapping and generating function

For clarity, here we consider the multidimensional vectors $[q,p]$, where $q, p \in \mathbb{R}^n$. Corresponding to the $2n$ Hamiltonian system, $q$ is the position variable, $q_i$ is the conjugate momentum variable. Then the standard symplectic form is defined to be

$$\Omega = dq \wedge dp. \tag{8}$$

where the operator $\wedge$ is called the exterior product or wedge product. Let $Q$ and $P$ be new variables, and we now investigate substitutions of the form

$$Q = Q(q, p), \quad P = P(q, p). \tag{9}$$

The change of variables in equation (9) is volume-preserving if and only if

$$dq \wedge dp = dQ \wedge dP, \tag{10}$$

which actually indicates that the transformation $(q, p) \rightarrow (Q, P)$ is symplectic. By equation (10), we can easily have

$$dp \wedge dq + dQ \wedge dP = 0. \tag{11}$$

This is sufficient to make sure the form $p dq + Q dp$ is exact, i.e. if we can find a function $F$ that satisfies

$$dF(q, p) = pdq + Q dp. \tag{12}$$

Since the differential form of $F$ can be written as

$$dF = \frac{\partial F}{\partial q} dq + \frac{\partial F}{\partial p} dp, \tag{13}$$

we can obtain the change of variables from $(q, p)$ to $(Q, P)$, by

$$p = \frac{\partial F}{\partial q}(q, p), \quad Q = \frac{\partial F}{\partial p}(q, p). \tag{14}$$

By the implicit function theorem, when the Hessian matrix $\frac{\partial^2 F}{\partial q \partial p}$ is non-singular, the equations in (14) are solvable for $P$ as a function of $q$ and $p$ and for $p$ as a function of $Q$ and $P$. So equation (14) gives a straightforward way to construct the symplectic transformation, and the function $F$ is usually called generating function. For more details, the reader is referred to Chapter 6 in [Meyer, Hall & Ollin (2009)].

### 2.3 Leap-frog neural network (LFNN)

In [Chen & Tao (2021)], the authors designed the network GFNN to learn the generating function, in order to obtain a symplectic mapping from one time point to the next. Then step by step, the system can continue to evolve with time. More precisely, for the one time step evolution, the mapping can be described as:

$$(q_0, p_0) \rightarrow (Q_1, P_1),$$

then denoting

$$\tilde{q}_1 = Q_1, \quad \tilde{p}_1 = P_1,$$

the next step is

$$(\tilde{q}_1, \tilde{p}_1) \rightarrow (Q_2, P_2).$$

This process could go on until

$$(\tilde{q}_{N-1}, \tilde{p}_{N-1}) \rightarrow (Q_N, P_N).$$

Here, the subscript $i$ corresponds to the $i$-th time point $t_i$ in equation (7). We note that, when $N$ is large, i.e. for the long time evolution, the above prediction of the time series could cause severe accumulative error.

Suppose we have the training set containing a time series in the interval $[t_0, t_{99}]$, i.e. for $N = 99$, with a fixed time
step of $h$. Different from [Chen & Tao, 2021], now we consider a finite partition of this time series, which is adopted to be 4 here. Thus we would get 4 subsets of the time series $(q_i, p_i)$:

- Partition 0 : $i = 0, \ldots, 24$, i.e. $t \in [t_0, t_{24}]$.
- Partition 1 : $i = 25, \ldots, 49$, i.e. $t \in [t_{25}, t_{49}]$.
- Partition 2 : $i = 50, \ldots, 74$, i.e. $t \in [t_{50}, t_{74}]$.
- Partition 3 : $i = 75, \ldots, 99$, i.e. $t \in [t_{75}, t_{99}]$.

Our new idea is to learn the generating function $F$ from the respective partitioned data, then $F$ is used to predict the time series. This process is visualized in Figs. 1 and 2. Let Partition 0 be the initial condition, we first study the generating function $F^1$ between Partition 0 and Partition 1. When we have $F^1(q, p) = q \cdot P + S^1(q, P)$, and $S^1$ is non-singular, we can learn $S^1$ by the neural network with respect to the loss function:

$$L_{S^1} = \frac{1}{25} \sum_{i=0}^{24} (||\partial_P S^1(q, P) - (Q - q)||^2 + ||\partial_Q S^1(q, P) - (P - p)||^2).$$

Then, Partition 1 can be generated by:

$$\begin{cases} p = P + \partial_0 S^1(q, P) \\ Q = q + \partial_P S^1(q, P) \end{cases},$$

i.e.

$$\begin{cases} p_i = p_{i+25} + \partial_q S^1(q_{i+25}, p_{i+25}) \\ q_{i+25} = q_i + \partial_p S^1(q_{i+25}, p_{i+25}), \end{cases}, i = 0, \ldots, 24,$$

and accordingly we would obtain the symplectic mappings

$$(q_0, p_0) \rightarrow (q_{25}, p_{25}), \ldots, (q_{24}, p_{24}) \rightarrow (q_{49}, p_{49}).$$

If the generating function $S^1$ is approximated with error $\epsilon$ in the first derivative, by the designed ANN, the $L_2$ norm of the deviation between the predicted sequence $\{(q_{25}, p_{25}), \ldots, (q_{49}, p_{49})\}$ and the ground truth sequence $\{(q_{25}, p_{25}), \ldots, (q_{49}, p_{49})\}$ satisfies, for any $i \in [25, 49],$

$$||p_i - \hat{p}_i||_2 \leq C \cdot \epsilon,$$

and

$$||q_i - \hat{q}_i||_2 \leq C \cdot \epsilon,$$

where $C$ is some constant.

It should be noted that, if we are predicting Partition 1 by the method we described above, we need to know the momenta $p_{25}, \ldots, p_{49}$. Since we do not have the ground truth for $p_i$ ($i \geq 25$), only the approximate values $\hat{p}_{25}, \ldots, \hat{p}_{49}$ could be estimated. In general, there are two ways to obtain them. The first one is that, starting with the initial condition Partition 0 : $(q_0, p_0), \ldots, (q_{24}, p_{24})$, we use the Newton’s method to solve the following nonlinear equations:

$$\hat{p}_{25} = p_0 - \partial_{q_{25}} S^1(q_0, \hat{p}_{25}),$$

$$\vdots$$

$$\hat{p}_{49} = p_{24} - \partial_{q_{24}} S^1(q_{24}, \hat{p}_{49}).$$

By this procedure we could get $\hat{p}_{25}, \ldots, \hat{p}_{49}$. The other way is more direct, as we can just train an additional fully connected neural network to predict these approximate momenta by using the supervised learning like we did in [Li et al., 2022].

Exactly the same, the Partition 2 and Partition 3 can
also be predicted from Partition 0, respectively, by introducing the individual generating functions $F^2(q, P) = q \cdot P + S^2(q, P)$ and $F^3(q, P) = q \cdot P + S^3(q, P)$. Then we could obtain the remaining two groups of symplectic mappings

$$\{q_0, p_0\} \rightarrow \{q_{50}, p_{50}\}, \ldots, \{q_{24}, p_{24}\} \rightarrow \{q_{74}, p_{74}\}. \quad (16)$$

and

$$\{q_0, p_0\} \rightarrow \{q_{75}, p_{75}\}, \ldots, \{q_{24}, p_{24}\} \rightarrow \{q_{99}, p_{99}\}. \quad (17)$$

In summary, for Partition 1, 2 and 3, if the respect generating functions $S^1, S^2$ and $S^3$ are approximated with errors $\leq \varepsilon$, the loss between the ground truth and the prediction satisfies the following inequalities:

$$\|p_i - \hat{p}_i\|_2 \leq C \cdot \varepsilon,$$

and

$$\|q_i - \hat{q}_i\|_2 \leq C \cdot \varepsilon,$$

(18)

where $i \in [25, 99]$.

The learning scheme designed above would be called the leap-frog neural network (LFNN). The main advantage of training the LFNN is that, we may avoid the issue of accumulation error since the symplectic mappings (15), (16) and (17) actually have no iteration, as we will show in the Section 3 by comparing our results with Chen & Tao (2021). Besides, the global error is rather small and the predictions of the time series in more general cases seem quite accurate.

### 2.4 Global Error formula for prediction

The considered sequence $(q_i, p_i)$ has 100 time points in total (i.e. $i = 0$-99), and is divided into 4 partitions. Thus, for each of the mappings (15), (16) and (17), the LFNN model is predicting a time series with a length of 25, e.g. $i = 25$-49 for Partition 1; in addition, since we will later consider the planar dynamical models, either $q_i$ or $p_i$ has two components.

Then for each of the partitions $(j = 1, 2, 3)$, we can define two $25 \times 2$ matrices as:

$$Q_{sim}^j(25 \times 2) = \{q_{25-j}, q_{25-j+1}, \ldots, q_{25-j+24}\},$$

$$P_{sim}^j(25 \times 2) = \{p_{25-j}, p_{25-j+1}, \ldots, p_{25-j+24}\},$$

where the subscript “sim” denote the ground truth data from numerical simulations. Similarly, we can also define the matrices predicted by the LFNN as:

$$Q_{LFNN}^j(25 \times 2) = \{\hat{q}_{25-j}, \hat{q}_{25-j+1}, \ldots, \hat{q}_{25-j+24}\},$$

$$P_{LFNN}^j(25 \times 2) = \{\hat{p}_{25-j}, \hat{p}_{25-j+1}, \ldots, \hat{p}_{25-j+24}\}.$$

If we have the $25 \times 2$ matrices $A$ and $B$, the distance between them can be defined as:

$$D(A, B) = \frac{1}{25} \sum_{k=1}^{25} \sqrt{\sum_{l=1}^{2}(A_{kl} - B_{kl})^2}.$$

Then, for a specific partition, we can calculate the average loss between prediction ($Q_{LFNN}^j$ and $P_{LFNN}^j$) and the ground truth ($Q_{sim}^j$ and $P_{sim}^j$). When we consider just one trajectory, the averaged losses are:

$$\text{loss}(Q^j) = D(Q_{LFNN}^j(25 \times 2), Q_{sim}^j(25 \times 2)),$$

$$\text{loss}(P^j) = D(P_{LFNN}^j(25 \times 2), P_{sim}^j(25 \times 2)). \quad (19)$$

And when we consider $M$ trajectories, the average losses are:

$$\text{LOSS}(Q^j) = \frac{\sum_{m=1}^{M} \text{loss}(Q_{m}^j)}{M},$$

$$\text{LOSS}(P^j) = \frac{\sum_{m=1}^{M} \text{loss}(P_{m}^j)}{M}. \quad (20)$$

Hereafter, without the superscript $j$, we refer to the loss calculated for all the three partitions, i.e. $i = 25$-99.

The said losses will be monitored on both the training and validation sets, and the the hyperparameters of our LFNN would be refined to make sure the global errors can decrease to small values. As a results, the precision of prediction may reach an acceptable level. In the next section, we will evaluate the prediction errors for Partitions 1, 2 and 3, respectively.

### 3 LEARNING IMPROVEMENT

Having desiging the LFNN, we now apply this network to a dynamical model considered in Chen & Tao (2021), i.e. the planar circular restricted 3-body problem (PCR3BP). And we expect to see the improvement on the GFNN for learning the motion of this system.

In the framework of PCR3BP, the primary $m_1$ and the secondary $m_2$ are moving on circular orbits about their common centre of mass $O$. These two massive bodies exert gravitational forces on the third massless particle but they are not affected by the particle. The motions of all these three bodies take place in the same plane. It is customary to choose non-dimensional parameters: the total mass of the system $(m_1 + m_2) = 1$; and for the primary and secondary, the mutual distance $a_N = 1$, and their angular velocity $\nu_N = 1$.

Considering the synodic coordinate system $(x, y)$, which has the origin located at $O$ but is rotating at a uniform rate $\nu_N$ in the counterclockwise direction, then the primary and secondary always lie along on the $x$-axis at $(-\mu, 0)$ and $(1 - \mu, 0)$, respectively. Here, the mass ratio $\mu = m_2/(m_1 + m_2)$ is the only adjustable parameter in the PCR3BP, e.g. for the system of Earth-Moon, Sun-planet or binary stars. Within one of these systems, the equations of the motion of the third massless particle can be written as

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x},$$

$$\ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y}, \quad (21)$$

where the “pseudo-potential” $U = U(x, y)$ is

$$U = \frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2}, \quad (22)$$

and $r_1$ and $r_2$ are the particle’s distances to the primary and secondary, respectively:

$$r_1^2 = (x + \mu)^2 + y^2,$$

$$r_2^2 = (x + \mu - 1)^2 + y^2. \quad (23)$$
It is well known that there is only one integral of the differential system \(\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, t)\), i.e. the Jacobi integral
\[
C(x, y, \dot{x}, \dot{y}) = 2U(x, y) - (\dot{x}^2 + \dot{y}^2),
\]
which is equivalent to the Hamiltonian quality of the PCR3BP. The preservation of \(C\) guarantees that the orbital evolution of the particle is symplectic.

We briefly review the settings of the 3-body system in Chen & Tao (2021) for complement: (1) the primary and secondary have the equal masses, i.e., \(\mu = 0.5\), and they form a binary; (2) for the particle, let \(\mathbf{q} = (x, y)\) be the position and \(\mathbf{p} = (\dot{x}, \dot{y})\) be the corresponding conjugate momentum; (3) for an orbit starting with \(\mathbf{q} = (5, 0)\) and \(\mathbf{p} = (0, 1)\), a number of time points have been recorded. Then these time points are adopted to be the initial conditions \(\mathbf{q}_0\) and \(\mathbf{p}_0\) to build the orbit database for the machine learning. In this way, \(10^5\) and \(10^6\) orbits are generated for the training set and the validation set, respectively; these ground truth or triple have the equal masses, i.e., \(\mu \approx 0.5\), which form a binary; (2) for the particle, let \(\mathbf{q} = (x, y)\) be the position and \(\mathbf{p} = (\dot{x}, \dot{y})\) be the corresponding conjugate momentum; (3) for an orbit starting with \(\mathbf{q} = (5, 0)\) and \(\mathbf{p} = (0, 1)\), a number of time points have been recorded. Then these time points are adopted to be the initial conditions \(\mathbf{q}_0\) and \(\mathbf{p}_0\) to build the orbit database for the machine learning. In this way, \(10^5\) and \(10^6\) orbits are generated for the training set and the validation set, respectively; these ground truth orbits are numerical calculated by the 4th order Runge-Kutta integrator with a step size of \(10^{-3}\). (4) for each orbit in the training and validation sets, a step of \(h = 0.1\) between two successive time points is adopted.

We notice that the two massive bodies locates at \((-0.5, 0)\) and \((0.5, 0)\), i.e. the distances to \(O\) are 0.5, while the particle is about ten times farther away from \(O\). Then such a hierarchical triple system can be deemed to an approximate 2-body system, where the particle is orbiting the center of mass \(O\). This implies that the evolution of the particle is in the non-chaotic regime of PCR3BP, and the regular motion could be easily learned by the neural network.

All of our experiments are taken in PYTORCH on a desktop CPU workstation. Similar to Chen & Tao (2021), in order to learn the generating function, we construct the LFNN consists of 6 fully connected layers. There are respectively 200, 100, 50, 20 neurons in each hidden layers, where the activation function tanch is used. We set the batch size to be 200, and the Adam optimizer is applied.

As described in Section 2.3, we split the sequence \((q_i, p_i)\) \((i \in [0, 99])\) into 4 partitions. Given the initial condition in the time interval \(t \in [t_0, t_{24}]\) (i.e. Partition 0), we are going to compare our results in the later evolution time of \(t \in [t_{25}, t_{99}]\) (i.e. Partitions 1, 2 and 3) with Chen & Tao (2021). We first make the prediction on Partition 1, then the generating function \(S^3\) has been learned, we would apply Newton’s method for
\[
P + \partial_q S^1(q, P) - p = 0.
\]
Let \(\mathbf{p} = \mathbf{p}_0\) and \(\mathbf{P} = \mathbf{p}_{25}\), and the initial point of the iteration \(\mathbf{p}_{24}\) be \(\mathbf{p}_{24}\), we can get \(\mathbf{p}_{25}\). Then in the formula
\[
Q = q + \partial_{\mathbf{p}} S^1(q, P),
\]
when we set \(q = q_0\), then we can have \(\mathbf{q}_{25} = \mathbf{Q}\). In the next step, let \(\mathbf{p} = \mathbf{p}_1\) and \(q = q_1\), we can obtain \(\mathbf{P} = \mathbf{p}_{26}\) and \(\mathbf{Q} = \mathbf{q}_{26}\); here the initial point for Newton’s method is taken to be \(\mathbf{p}_{25}\). This process continues until \(i = 49\), and the prediction on Partition 1 is achieved.

Independently, we then consider the prediction on Partition 2. Also assume Partition 0 to be initial data, we repeat the above process to generate the symplectic mappings. Similarly, the symplectic mappings for Partition 3 can be obtained as well.

After finishing the network training, we pick the same test orbit with \(q_0 = (5.01, -0.01), p_0 = (0, \frac{1}{\sqrt{5}})\) as considered in Chen & Tao (2021). For this test orbit, the averaged losses of \(Q^f\) from both our best-trained LFNN and the GFNN are presented in Table 1.

It is obvious that, our method has quite small losses among all the 3 partitions; and most importantly, these losses corresponding to different periods of the orbit prediction are nearly the same, around 0.003-0.004. For the early evolution on Partition 1, i.e. \(t \in [t_{25}, t_{49}]\), the GFNN has the comparable loss to our results, about 0.0035. While for the later evolution on Partition 2, i.e. \(t \in [t_{50}, t_{74}]\), the GFNN’s loss increases to about 0.0050, which is \(\approx 43\) per cent larger. And, for the end of the evolution on Partition 3, i.e. \(t \in [t_{75}, t_{99}]\), this loss continues to increase, up to a much larger value of 0.0079. From this point of view, the error of GFNN’s prediction seems to grow with time, while our LFNN’s prediction has nearly a stationary error. Thus our results show an advantage to predicting the long time evolution via the multiple-steps learning.

### Table 1

| Partition j | Time interval | Network  | loss(Q^f) |
|-------------|---------------|----------|-----------|
| j = 1       | [t_{25}, t_{49}] | GFNN     | 0.003530  |
|             |               | LFNN     | 0.003334  |
| j = 2       | [t_{50}, t_{74}] | GFNN     | 0.005018  |
|             |               | LFNN     | 0.004090  |
| j = 3       | [t_{75}, t_{99}] | GFNN     | 0.007859  |
|             |               | LFNN     | 0.003086  |

4 APPLICATIONS

Beyond the orbit of Neptune, there exists a disk of small bodies orbiting the Sun, known as the Kuiper belt. By now, thousands of the Kuiper belt objects (KBOs) have been observed, and they can be classified by their orbital characteristics: the resonant, the classical (i.e. non-resonant), the scattered, and the detached objects. The studies of the KBOs’ dynamical evolution may provide important clues to the early history of our Solar system.

We now intend to use the designed LFNN to learn the behavior of the KBOs outside or in the mean motion resonance (MMR) with Neptune. In the framework of the PCR3BP, we refer to the Sun with mass \(m_1\) as the primary, the planet Neptune with mass \(m_2\) as the secondary, and a KBO as the third body. So in the equations of the motion of the KBO particle, i.e. equation \(21\), the mass ratio \(\mu \approx 5.146 \times 10^{-5}\). Then we can generate the training and validation data by numerical simulations. Still, we take 100
time points for each KBO’s orbit, for a certain evolution time.

4.1 Non-resonant KBOs

The classical KBOs have semimajor axes roughly between 42 AU to 48 AU, and are defined to be objects outside Neptune’s MMRs. In the PCR3BP, we consider the particles having initial non-dimensional semimajor axes $a = 1.44$. Since the distance between the Sun and Neptune, i.e., $\sim 30$ AU, is set to be 1, the particles’ actually semimajor axes can be calculated to be about 43.2 AU. At this location, none of the MMRs with order lower than 8th exists. Besides, the KBO particles are placed on the nearly circular orbits with eccentricities $e = 0.01$, to additionally make sure that they are non-resonant.

As in Section 3.1, the ground truth orbits are generated by using the 4th order Runge-Kutta integrator, and the the numbers of the orbits in the training and validation sets are $10^3$ and $10^2$, respectively. For the 100 time points with a step of $\tau = 0.1$, the total evolution time is 10 in the non-dimensional unit. According to Kepler’s third law, the orbital period of a particle is $\sqrt{\frac{a^3}{G\mu}} \approx 1.441 \times 2\pi \approx 7.1$. It suggests that the particles complete more than 1 full orbit over the evolution time of 10.

For the said particles starting with $a = 1.44$ and $e = 0.01$, their initial longitudes of perihelia $\varpi$ are set to be 60°, while the mean longitudes $\lambda$ are randomly selected between 0 and 2$\pi$. Given such initial conditions, we generate the training and validation samples, which are out of the resonance regions.

Then we are about to re-trained our best LFNN model built on the generating function. Bearing in mind that, in order to make the symplectic transformations via the obtained generating function $S(q, P)$, we need to know the new momentum $\tilde{P}$, i.e., $\{\tilde{p}_{25}, \ldots, \tilde{p}_{31}\}$. If we apply the Newton’s method to the 100 validation orbits, in order to get the losses of $P$ and $Q$, it is not easy to find the proper values of $P$. This difficulty comes from the fact that, when we adopt the leap-frog learning, the gradient of the generating function could be not accurate enough. As we mentioned in Section 2.3, alternatively we could train an additional supervised neural network to learn the approximate $P$. Then by $Q = q + \partial PS(q, P)$, the new position $Q$ can be predicted.

In Fig. 3 for each of the 100 trajectories from the validation set, we plot averaged losses of the predicted positions $\tilde{q}$ (i.e. loss($Q$), see blue dots) and momenta $\tilde{p}$ (i.e. loss($P$), see yellow dots). We find that the maximal loss is less than 0.0009, which is associated to $\tilde{q}$. We further notice that the losses of $\tilde{p}$ are generally even smaller, on the order of loss($Q$) $\sim$ 0.0001-0.0005. Comparing with the case of “binary + particle” in Section 3.1, where loss($Q$) is an order of magnitude larger, the LFNN seems to work better on learning the evolution of non-resonant KBOs.

4.2 Resonant KBOs

The population of resonant KBOs is believed to be a solid evidence for the orbital migration of Jovian planets. The study of the resonant KBOs is very important for us to understand the early shake-up of the outer Solar system, and a lot of works have been carried out even since early 1990s [Malhotra, 1993, 1995, Hahn & Malhotra, 2000, Li, Zhou & Sun, 2014, Nesvorný & Vokrouhlický, 2016, Pike & Lawler, 2017, Lawler et al., 2019]. For this reason, in [Li et al., 2022] we provide a machine learning method that can fast predict the orbital evolution of the KBOs in the 2:3 MMR with Neptune. In the framework of the PCR3BR, since the trajectories of the 2:3 resonators are restricted on the invariant tori, their motions are regular and were learned rather accurately.

As we stated in Section 1, we intended to use the position $q$ and momentum $p$ as training data because the same order of magnitude of these variables could be more suitable for the machine learning. However, for $q$ and $p$, our designed ANN in [Li et al., 2022] can not keep the system symplectic by conserving the Jacobi integral. Now we would test how the new LFNN performs on predicting the time series $(q_i, p_i)$.

For the samples particles, their initial conditions are exactly the same as we used in [Li et al., 2022]: $a_0 = 1.31$ (the location of the nominal 2:3 resonance), $e_0 = 0.1$ (for regular motion everywhere in the phase space), $\varpi_0 = 60^\circ$ (unchanged by rotating the coordinate system); and, the initial resonant angles $\sigma_0$ are chosen in the region of $[180^\circ - 120^\circ, 180^\circ + 120^\circ]$, where $180^\circ$ is the stable libration centre. Then the initial value of $\lambda$ can be determined from a given $\sigma$ by

$$\sigma = 3\lambda - 2\lambda_N - \varpi, \tag{27}$$

where $\lambda_N$ is the mean longitude of Neptune. For a given set of the orbital elements $(a, e, \varpi, \lambda)$, the initial conditions of the resonators are generated in terms of the position $x$ and velocity $\nu$. Then we numerically propagate the equations of motion, i.e., equation (21), by adopting the 8th-order Runge-Kutta integrator with a time step of $\sim 0.01$ yr and the local error tolerance of $10^{-20}$.

In order to learn the behaviour of the resonant KBOs, the time evolution should cover at least a full libration cycle. For the 2:3 resonators, the libration period is about 20000
Here, the horizontal axis shows the initial resonant angle $\sigma_0$ from $60^\circ$ to $300^\circ$, centered at the $180^\circ$ resonant centre.

For a quick summary, we address that the LFNN works sample “tp1” starting with $\sigma_0 = 60^\circ$, which has the largest loss of loss$(Q \sim 0.08)$ and loss$(P \sim 0.05)$ (see Fig. [4]). Then we present in Fig. [5] the evolution for this sample from the validation set, and the associated orbital parameters can be simply obtained from the position and velocity variables. The black dots represent the ground truth values calculated by numerical integration. While the red dots denote the prediction made by the LFNN for the time period from $t_{25}$ to $t_{99}$, where $t_{99}$ corresponds to the real time of 25000 yr. As a comparison, we apply the fully connected ANN in [Li et al. 2022] to an additional prediction, and the outputs of this ANN are indicated by the green dots.

The upper panel of Fig. 5 shows the time evolution of the Jacobi integral $C$ for the sample tp1. It is quite clear that the value of $C$ deduced from our previous method (green dots) seem to considerably depart from the ground truth (black dots), indicating that the conservation of $C$ can hardly hold. But, with regard to our new LFNN, the approximate $C$ values (red dots) are different from the ground truth values only by less than 0.01 at any time point. Therefore, the first improvement of the LFNN on predicting the motion of the 2:3 resonators is that, the symplecticity of the orbital evolution can be well maintained.

As shown in the lower four panels of Fig. 5, the second improvement is the increased accuracy on the predictions of the orbital elements $a$, $e$, $\varpi$ and $M$ ($M = \lambda - \varpi$ is the mean anomaly). We can see that, the red dots are much closer to the black dots representing the ground truth than the green dots, especially for the slow variables $a$ and $e$. In this sense, the designed LFNN can help to significantly reduce the errors for making the long time prediction, and thus is better than the simple fully connected ANN.

In a brief summary, we address that the LFNN works well even when the step $h$ of the time series $(q_i, p_i)$ is relatively large.

5 CONCLUSIONS

To improve our previous study of machine learning prediction for the 3-body problem, in which we considered a case of regular motion in the Hamiltonian system [Li et al. 2022], this paper is devoted to maintain the symplecticity of the system’s evolution. In the framework of PCR3BP, [Li et al. 2022] tried to train a fully connected ANN to directly learn the generated function. In mathematics, the method of generating function can naturally lead to the symplectic transformation from one time point to another. As an improvement, we further split the 100 time points into 4 partitions, i.e. Partition 0: $(q_0, p_0), \ldots, (q_{24}, p_{24})$, Partition 1: $(q_{25}, p_{25}), \ldots, (q_{50}, p_{50})$, Partition 2: $(q_{51}, p_{51}), \ldots, (q_{74}, p_{74})$, and Partition 3: $(q_{75}, p_{75}), \ldots, (q_{99}, p_{99})$. Then we build the symplectic evolution maps, from the initial condition of Partition 0 to each of Partitions 1, 2 and 3, respectively by the generating
Figure 5. For the validation sample tp1 starting with $\sigma_0 = 60^\circ$ (see Fig. [1]), the temporal evolution of Jacobi integral (upper panel) and the orbital elements $a$, $e$, $\varpi$, $M$ (lower four panels). The black dots refer to the ground truth calculated by the numerical simulations. Provided the first 1/4 period of the ground truth orbit as the initial condition, the following 3/4 period data is respectively predicted by the new LFNN (red dots) and the fully connected ANN in Li et al. (2022) (green dots). As in the framework of PCR3BP, the semimajor axis values, $a$, are in respect with Neptune’s semimajor axis $a_N \sim 30$ AU.

functions $S^1$, $S^2$ and $S^3$. When the generating functions are approximated by the neural network, we can get three independent groups of the flow maps (15), (16) and (17). This new structure of machine learning is named leap-frog neural network (LFNN).

The designed LFNN is first applied to the same PCR3BP model considered in Chen & Tao (2021), i.e. a massless particle orbiting around a binary with equal masses. For learning the time series ($q_i$, $p_i$), we adopt the same step of $h = 0.1$ as used by the GFNN. The results show
that, unlike the prediction of the GFNN, which has an error growing with time, our LFNN’s prediction has a stationary error at rather a small value. We then argue that the LFNN could avoid the issue of accumulative error, and thus may be helpful to increase the accuracy for making the long time prediction. This serves as the main advantage of our LFNN.

As an application of the designed LFNN, we have evaluated the prediction on the behavior of the KBOs beyond the orbit of Neptune. For the non-resonant KBOs, their orbital periods are about 7.1 in the non-dimensional unit. Thus, as in the case of “binary + particle”, a small time step of \( h = 0.1 \) for the 100 time points is also adopted to train the network, leading to the evolution time of 10 which is longer than the said period of 1 full orbit. As expected, such a \( h \) can result in very accurate prediction, with the loss on the order of \( \mathcal{O}(0.0001) \).

But for the 2:3 resonant KBOs, we need to know the evolution over at least a full libration cycle of about 20000 yr. Thus we increase the time step to \( h = 9.65 \), which leads to a total evolution time of 25000 yr as we used in [Li et al. (2022)]. Although this \( h \) becomes almost 100 time larger, our LFNN can still well work. By comparing with prediction of the ANN used in [Li et al. (2022)], the LFNN has two significant improvements: (1) the predicted orbits could conserve the Jacobi integral at quite a high level; (2) the predictions of the orbital elements \( a, e, \varpi \) and \( M \) are much more accurate, even for the validation sample with maximum loss. These results supports our argument that the LFNN may be useful to learn the long time evolution of the Hamiltonian system.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Nos. 11973027, 11933001 and 11601159), and National Key R&D Program of China (2019YFA0706601).

DATA AVAILABILITY

The data underlying this article are available in the article and in its online supplementary material.

REFERENCES

Agarwal S., Davé R., Bassett B. A., 2018, MNRAS, 478, 3410
Ahmed N., et al., 2010, Econometric Reviews, 29(5-6):594-621.
Armstrong D. J., Gamper J., Damoulas T., 2021, MNRAS, 504, 5327
Baqui P. O. et al., 2021, A&A, 645, A87
Bertalan, T., et al., 2019, Journal of Nonlinear Science, 29(12): 121107
Box G, Jenkins G., Time Series Analysis: Forecasting and Control. Holden-Day; 1976.
Breen P. G., et al., 2020, MNRAS, 494, 2465
Chen, Z., et al., 2020, In International Conference on Learning Representations (ICLR), preprint arXiv:1909.13334

Chen R., Tao M., 2021, in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July
E W., 2017, Communications in Mathematics and Statistics, 5(1):1-11;
Gardner J., 1985, Journal of Forecasting, 4(1):1-28.
Gladden B., Marsden B. G., Vanlaerhoven C., 2008, in Barucci M. A., Boehnhardt H., Cruikshank D. P., Morbidelli A., eds, Nomenclature in the Outer Solar System, In The Solar System Beyond Neptune, University of Arizona Press, Tucson, p. 43
Gallardo T., 2006, Icarus, 184, 29
Greydanus S., Dzamba M., Yosinski J., 2019, preprint (arXiv:1906.01563v3)
Harvey A., 1990, Forecasting, Structural Time Series Models and the Kalman Filter, Cambridge University Press.
Hahn J. M., Malhotra R., 2005, AJ, 130, 2392
Jin, P., et al., 2020, Neural Networks, 132:166-179
Khain T. et al., 2020, AJ, 159, 133
Koon W., et al., 2020, Chaos: An Interdisciplinary Journal of Nonlinear Science, 10(2): 427-469,
Li J., Zhou L.-Y., Sun Y.-S., 2014, MNRAS, 437, 215
Lawler et al., 2019, AJ, 157, 253
Li G., Holman M. & Tao M., 2016, 831(1): 96
Li X., et al., 2022, MNRAS, 511, 2218-2228
Lutter M., Ritter C., and Peters J., 2019, In International Conference on Learning Representations
Lucic-Smith M., Peiris H. V., Pontzen A., 2019, MNRAS, 490, 331
Meyer K., Hall G., Offin D.: 2009, Introduction to Hamiltonian dynamical systems and the N-body problem, 2nd Edition, Springer, Berlin,p45-63
Malhotra R., 1993, Nature, 365, 819
Malhotra R., 1995, AJ, 110, 420
McCulloch W. S., Pitts W., 1943, The Bulletin of Mathematical Biophysics, 5, 115
Nesvorný D., Vokrouhlický D., 2016, ApJ, 825, 94
Petulante A. et al., 2021, MNRAS, 504, 248
Pike R. E., Lawler S. M., 2017, AJ, 154, 171
Poincaré H., 1892, Les methodes nouvelles de la mecanique celeste, Gauthier Villars, Paris, Vols. 1-3
Quarles B., et al., 2020, The Astronomical Journal, 159(3):80,
Rosenblatt F., 1958, Psychol. Rev., 65, 386
Schanche N. et al., 2019, MNRAS, 483, 5534
Toth P., et al., 2020, In International Conference on Learning Representations (ICLR)
Winters P., 1960, Management Science, 6(3):324-342.
Wu K., Qin T., and Xiu D., SIAM Journal on Scientific Computing, 42(6): A3704-A3729.
Xiong S., et al., 2021, In International Conference on Learning Representations (ICLR)
Vavilova I. B. et al., 2021, A&A, 648, A122
Zhang K. et al., 2019, ApJ, 883, 63
Zhong Y., Dey B., and Chakraborty, A.,2020, International Conference on Learning Representations(ICLR)