UNIVERSAL APPROXIMATION PROPERTIES OF NEURAL NETWORKS FOR ENERGY-BASED PHYSICAL SYSTEMS

Yuhan Chen
Graduate School of System and Information Science
Kobe University
Hyogo, Japan
193x226x@stu.kobe-u.ac.jp

Takashi Matsubara
Graduate School of Engineering Science
Osaka University
Osaka, Japan
matsubara@sys.es.osaka-u.ac.jp

Takaharu Yaguchi
Graduate School of System and Information Science
Kobe University
Hyogo, Japan
yaguchi@pearl.kobe-u.ac.jp

February 25, 2021

ABSTRACT

In Hamiltonian mechanics and the Landau theory, many physical phenomena are modeled using energy. In this paper, we prove the universal approximation property of neural network models for such physical phenomena. We also discuss behaviors of the models for integrable Hamiltonian systems when the loss function does not vanish completely by applying the KAM theory.

1 Introduction

Many physical phenomena are described by energy-based theories such as Hamiltonian mechanics and the Landau theory. In such theories, equations of the following form are often used as governing equations:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = G\frac{\partial H}{\partial u}, \tag{1}$$

where $u : t \in \mathbb{R} \mapsto u(t) \in \mathbb{R}^N$, $H : u \in \mathbb{R}^N \mapsto H(u) \in \mathbb{R}$, G is a skew-symmetric or negative semidefinite matrix Furihata & Matsuo (2010). H represents the energy function of the system. For the systems described by (1), the time derivative of the energy is given as

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\partial H}{\partial u}^\top \frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\partial H}{\partial u}^\top G\frac{\partial H}{\partial u}. \tag{2}$$

Hence, when the matrix G is skew-symmetric, the energy conservation law holds

$$\frac{\mathrm{d}H}{\mathrm{d}t} = 0$$

because for all vector v, $v^\top Gv = 0$ when G is skew-symmetric. In particular, if G is in addition nondegenerate, (4) is called Hamiltonian equation, which is the equation of motion in Hamiltonian mechanics, and the energy function H is called Hamiltonian (e.g., Marsden & Ratiu (2013)). When the matrix G is negative semidefinite, the energy function is monotonically non-increasing:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\partial H}{\partial u}^\top G\frac{\partial H}{\partial u} \leq 0. \tag{3}$$

The equations of this form are used in many fields of mathematical modeling, for example, in the Randau theory and the phase-field method. Phenomena such as phase separation, crystal growth and crack propagation are modeled by using these theories (e.g., Caginalp (1986); Wheeler et al. (1993); Steinbach (2009); Miehe et al. (2010)).

In recent years, there has been a lot of research on predicting the corresponding physical phenomena by learning the energy function H in such equations with a neural network H_{NN} (e.g., Greydanus et al. (2019); Cranmer et al. (2020); Chen et al. (2020); Zhong et al. (2020); Matsubara et al. (2020)); however, to the best of our knowledge, the approximation properties of such models have not been theoretically investigated sufficiently, except for the SympNet (Jin et al., 2020b) for the Hamilton equation, where the universal approximation theorems for discrete-time neural network models are provided.

In this paper, we present some theoretical analyses of the approximation properties of such a model, which are outlined in Figure 1. Firstly, we will show that the continuous-time energy-based deep physical model

$$\frac{\mathrm{d}u}{\mathrm{d}t} = G\frac{\partial H_{\mathrm{NN}}}{\partial u} \tag{4}$$

has the universal approximation property. This is obtained by applying the theory by Hornik et al. (1990) that shows the universal approximation property of multilayer feedforward networks for the function values along with the derivatives. Secondly, we show the universal approximation property for a coordinate transformation of the above model

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (\frac{\partial u}{\partial x})^{-1} G (\frac{\partial u}{\partial x})^{-\top} \frac{\partial H}{\partial x}. \tag{5}$$

Models of this form appear when the state variables in the original equation, especially the generalized momenta for Hamiltonian equations, are unknown, and hence they must be learned as latent variables by using, e.g., neural networks.

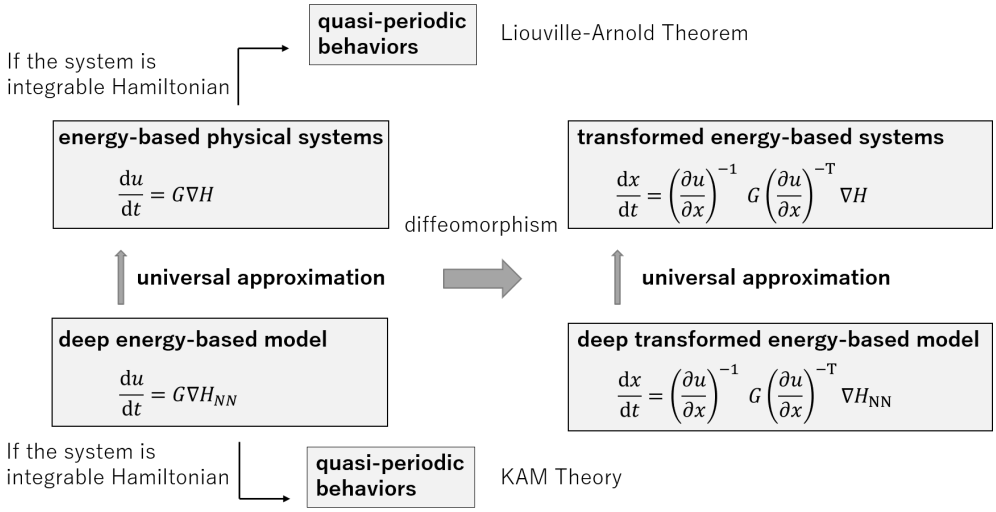


Figure 1: The outline of this paper. We show the universal approximation theorem for the general deep energy-based models with and without the latent variables. The analysis of the models for the integrable Hamiltonian systems are also provided; the neural network models admit the quasi-periodic behaviors of the original systems even when the loss function does not vanish but is small enough.

Thirdly, we consider the behavior of one of the most important energy-based models, Hamiltonian neural networks (Greydanus et al., 2019), especially when the loss function does not completely vanish, under the assumption that the learning target is an integrable system in the sense of Liouville. In this case, the trained model can be regarded as a perturbed Hamiltonian system due to the modeling error in the energy function, and hence in principle can be analyzed by applying the famous theory of mathematical physics, the Kolmogorov–Arnold–Moser (KAM) theory. When the original system is integrable the system exhibits quasi-periodic behaviors, which are, roughly, superpositions of periodic motions with mutually unrelated frequencies. The KAM theory shows that these quasi-periodic behaviors persist after sufficiently small perturbations; thus it can be shown that the trained model has the quasi-periodic behaviors even if the loss function does not vanish completely, provided that the loss function is small enough. We performed a rigorous analysis on this to find that *it would be important to minimize the p-norm of the errors with a sufficiently large p rather than the mean squared errors.* This intuitively makes sense because the KAM theory requires that the perturbation is small in the sup norm and the p-norm approximates this norm when p is large enough.

The main contributions include:

Universal approximation theorem for general energy-based neural network models with and without hidden variables While applications of neural networks to modeling of physical phenomena are developing very rapidly, as seen in Section 2, theoretical studies on these models are limited. This study aims to provide a theoretical basis for these approaches.

Application of the KAM theory to the analysis of the behaviors of Hamiltonian neural networks when the loss function does not vanish Even for a model with the universal approximation property, values of the loss function can not be completely vanished when the model is actually trained. In this paper, for integrable systems in the sense of Liouville, by combining the KAM theory and the generalization error analysis using the Rademacher complexities, we show that under a certain condition with a high probability quasi-periodic motions of such systems are preserved even when the loss function does not vanish. In particular, the analysis shown below implies that the p-norm of the errors should be minimized rather than the mean squared errors of them.

Introduction of general energy-based neural network models with hidden variables While the main purpose of this study is the theoretical analysis of deep physical modeling, the latent variable model considered in Section 4 is a new energy-based neural network model that does not depend on a specific coordinate system. For example, the Hamilton equation used in Hamiltonian neural networks requires the state of the system to be described in a special coordinate system with generalized coordinates and generalized momenta. However, the generalized momenta depend on the Lagrangian, which is defined by the unknown energy function modeled by the neural networks, of the system and hence is often unknown without detailed prior knowledge about the system. For the Hamilton equation, a model that employs latent variables has been proposed (Toth et al., 2019); however, the main objective of Toth et al. (2019) is the extraction of low-dimensional phase space from images or movies, while ours is finding a diffeomorphism.

2 Related work

Many studies of neural network models for phenomena that can be modeled by the energy-based equation (1) have been proposed. Among those studies, the most basic ones would be neural differential equations (Chen et al., 2018) and Hamiltonian neural networks (Greydanus et al., 2019). In particular, the extensions of Hamiltonian neural networks have been intensively developed.

Due to the limitation of space, it is difficult to describe all of them, but some examples are as follows: Toth et al. (2019) extended Hamiltonian neural networks to the models with latent variable models. Other studies such as Zhong et al. (2020), DiPietro et al. (2020), and Xiong et al. (2021) focus on the symplectic structure of the Hamilton equation. Focusing on Noether's theorem, which is a fundamental theorem of classical mechanics, several studies such as Bondesan & Lamacraft (2019), Finzi et al. (2020a), and Bharadwaj et al. (2020) have developed methods related to symmetry and conservation laws. In addition, Matsubara et al. (2020) constructed a discrete-time model that preserves the energy behaviors by using the same model as (4) in this paper. Besides, in Galioto & Gorodetsky (2020), Hamiltonian neural networks are combined with the Bayesian approach.

Methods applied to the framework of classical mechanics other than Hamiltonian mechanics include Cranmer et al. (2020), Desai & Roberts (2020), and Sæmundsson et al. (2019), which are methods for Lagrangian formalism. In Jin et al. (2020c), reinforcement learning is applied to the variational principle. Finzi et al. (2020b) proposed a simplified model by introducing constraints. In Jin et al. (2020a), Hamiltonian neural networks are extended to the Poisson system, which is a wider class of mechanical equations. There are also a number of proposals that integrate with more advanced deep learning techniques, e.g., graph networks (Sanchez-Gonzalez et al., 2019), recurrent neural networks (Chen et al., 2020) and normalizing flows (Li et al., 2020). As an application-oriented approach, Feng et al. (2020) designed the model for structural analysis from the microscopic level.

On the other hand, as far as the authors know, there is no theoretical research other than the universal approximation theorems for Hamiltonian mechanics in SympNet (Jin et al., 2020b), in which a certain kind of neural networks are shown to have universal approximation properties for symplectic maps. The difference between their results and ours is in that the universal approximation theorems in Jin et al. (2020b) are for discrete-time models while ours are for continuous-time models.

3 Universal approximation properties of neural network models for energy-based physical systems

In this section, we show that under a certain condition the model (4) has the universal approximation property.

We first prepare some notation to describe the theorem. We denote $C^m(X)$ with the topology of the Sobolev space $W^{p,m}(X)$ by $S_p^m(X)$. $W^{p,m}(X)$ is a space of functions that admit weak derivatives up to the mth order of which L^p norms are bounded; hence $S_p^m(X)$ is the space of functions in $W^{p,m}(X)$ with (usual) derivatives. For details on the Sobolev theory, see Adams & Fournier (2003). We denote L^p norms for functions by $\|\cdot\|_{L^p}$ and that for vectors by $\|\cdot\|_p$. The space of the neural networks with the activation function σ is denoted by $\Sigma(\sigma)$:

$$\Sigma(\sigma) = \{g : \mathbb{R}^r \to \mathbb{R} \mid g(x) = \sum_i^q \beta_i \sigma(\gamma_i^\top x + \alpha_i),\ \alpha_i \in \mathbb{R}, \beta_i \in \mathbb{R}, \gamma_i \in \mathbb{R}^r\}.$$

The following is the first result of this paper, which shows the universal approximation property of the model (4).

Theorem 1. *Let $H : \mathbb{R}^N \to \mathbb{R}$ be an energy function with the equation*

$$\frac{\partial u}{\partial t} = G\frac{\partial H}{\partial u},$$

where $u : t \in \mathbb{R} \mapsto u(t) \in \mathbb{R}^N$ and G is an $N \times N$ matrix. Suppose that the phase space K of this system is compact and the right-hand side $G\partial H/\partial u$ is Lipschitz continuous. If the activation function $\sigma \neq 0$ belongs to $S_2^1(\mathbb{R})$, then for any $\varepsilon > 0$ there exists a neural network H_{NN} for which

$$\left\| G\frac{\partial H}{\partial u} - G\frac{\partial H_{\mathrm{NN}}}{\partial u} \right\|_2 < \varepsilon$$

holds.

This theorem is a direct application of the universal approximation theorem of Hornik et al. (1990). In Hornik et al. (1990), several universal approximation theorems are provided, for example,

Theorem 2 (Hornik et al., 1990)**.** *Let the activation function $\sigma \neq 0$ belong to $S_p^m(\mathbb{R}, \lambda)$ for an integer $m \geq 0$. Then, $\Sigma(\sigma)$ is m-uniformly dense in $C^\infty(K)$, where K is any compact subset of $\mathbb{R}^N$.*

Lemma 1. *Under the same assumption, $\Sigma(\sigma)$ is also dense in $S_p^m(\mathbb{R}, \lambda)$.*

Hence if the activation function σ of the hidden layer is in $S_p^m(\mathbb{R}, \lambda)$ and does not vanish everywhere, then for any sufficiently smooth function there exists a neural network that approximates the function and its derivatives up to the order m arbitrarily well on compact sets. Besides, this theorem is also extended to the functions of multiple outputs; see Hornik et al. (1990).

Proof of Theorem 1. Since the target equation is determined only by the gradient of H, any function obtained by shifting H by a constant gives the same equation. Hence we choose and fix an energy function H that yields the target equation. Because $G\partial H/\partial u$ is Lipschitz continuous and hence continuous on the phase space K, this function is bounded and square-integrable. Thus $G\partial H/\partial u \in S_2^0(K)$, which means H is in $S_2^1(K)$. Therefore, from Lemma 1 and the assumption that the activation function is in $S_2^1(\mathbb{R})$, for each ε there exists a neural network that approximates H in $S_2^1(K)$:

$$\|H - H_{\mathrm{NN}}\|_2^2 + \left\| \frac{\partial H}{\partial u} - \frac{\partial H_{\mathrm{NN}}}{\partial u} \right\|_2^2 < \frac{\varepsilon^2}{\|G\|_2^2},$$

which gives

$$\left\| G\frac{\partial H}{\partial u} - G\frac{\partial H_{\mathrm{NN}}}{\partial u} \right\|_2^2 \leq \|G\|_2^2 \left\| \frac{\partial H}{\partial u} - \frac{\partial H_{\mathrm{NN}}}{\partial u} \right\|_2^2 < \varepsilon^2.$$

□

Remark 1. *In the above theorem, we supposed that the energy function is Lipschitz continuous; however, as can be seen from the proof, if the energy function is C^∞ the function can be approximated by a C^∞ neural network provided that the activation function is smooth enough. We use this statement later in Section 5.*

4 Universal approximation property of deep energy-based models with latent variables

The practical use of Hamiltonian neural networks is hampered by the fact that the state variables must be represented by using a specific coordinate, e.g., the generalized momentum. The generalized momenta are in principle defined by using the Lagrangian of the system; however, the Lagrangian is typically defined by the difference between the kinetic energy and the potential energy. Thus, the derivation of the generalized momenta requires the energy function, which is unknown and hence is the target function of Hamiltonian neural networks.

Therefore, we propose a model for physical phenomena that cannot be modeled by the normal form (4) due to the impossibility of calculating the required specific coordinate system but are considered to follow the energy-based physics theories. More specifically, we consider the equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) \tag{6}$$

that can be transformed into the model equation (4) with respect to the hidden state variable $u(t) = u(x(t))$ by a certain diffeomorphism.

By substituting $u = u(x)$ into the model equation (4), we have

$$\frac{\partial u}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} = G\frac{\partial x}{\partial u}^{\top}\frac{\partial H}{\partial x}.$$

Since $\partial x/\partial u = (\partial u/\partial x)^{-1}$, this becomes

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (\frac{\partial u}{\partial x})^{-1}G(\frac{\partial u}{\partial x})^{-\top}\frac{\partial H}{\partial x}. \tag{7}$$

In fact, the transformed model (7) also has the same energetic property as the original equation.

Theorem 3. *The model* (7) *admits the energy conservation law*

$$\frac{\mathrm{d}H}{\mathrm{d}t} = 0$$

if G is skew-symmetric and admits the energy dissipation law

$$\frac{\mathrm{d}H}{\mathrm{d}t} \leq 0$$

if G is negative semidefinite.

Proof. This theorem can be obtained from

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\partial H}{\partial x}^{\top}\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\partial H}{\partial x}^{\top}(\frac{\partial u}{\partial x})^{-1}G(\frac{\partial u}{\partial x})^{-\top}\frac{\partial H}{\partial x}$$

which is 0 if G is skew-symmetric and ≤ 0 if G is negative semidefinite. □

The second result in this study is the universal approximation property of the above transformed model (7).

Theorem 4. *Let $H : \mathbb{R}^N \to \mathbb{R}$ be an energy function for the equation*

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (\frac{\partial u}{\partial x})^{-1}G(\frac{\partial u}{\partial x})^{-\top}\frac{\partial H}{\partial x}$$

where $x : t \in \mathbb{R} \mapsto x(t) \in \mathbb{R}^N$, $u : x \in \mathbb{R}^N \mapsto u(x) \in \mathbb{R}^N$ and G is an $N \times N$ matrix. Suppose that the phase space K of this system is compact and the right-hand side $\partial H/\partial u$ is Lipschitz continuous. Suppose also that u is a C^1-diffeomorphism. If the functions $\sigma \neq 0$ and $\rho \neq 0$ belong to $S_2^1(\mathbb{R})$, then for any $\varepsilon > 0$ there exist neural networks H_{NN} with the activation function σ and u_{NN} with ρ for which

$$\left\|(\frac{\partial u}{\partial x})^{-1}G(\frac{\partial u}{\partial x})^{-\top}\frac{\partial H}{\partial x} - (\frac{\partial u_{\mathrm{NN}}}{\partial x})^{-1}G(\frac{\partial u_{\mathrm{NN}}}{\partial x})^{-\top}\frac{\partial H_{\mathrm{NN}}}{\partial x}\right\|_2 < \varepsilon$$

holds.

Proof. We need to prove the approximation property for $(\partial u/\partial x)^{-1}$. From the assumption that $\rho \neq 0$ is in $S_2^1(\mathbb{R})$, there exists a function u_{NN} that approximates $\partial u/\partial x$. Since the determinant function of matrices is continuous, it is deduced that $\det \partial u_{\mathrm{NN}}/\partial x \neq 0$ and hence $(\partial u_{\mathrm{NN}}/\partial x)^{-1}$ exists. Because the matrix inverse is also continuous, $(\partial u_{\mathrm{NN}}/\partial x)^{-1}$ is also approximated by u_{NN}. □

Remark 2. *Although the main objective of this study is the theoretical analysis, to the best of our knowledge, no deep learning model using this equation has been proposed so far.*

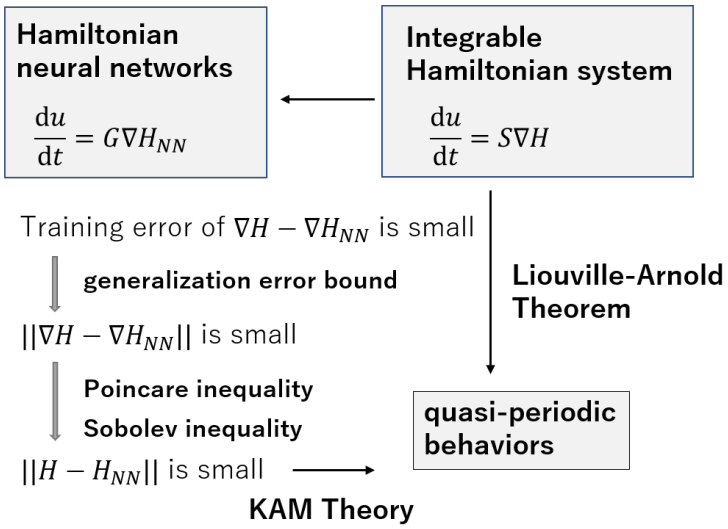


Figure 2: The outline of the anaylsis in Section 5. Integrable Hamiltonian systems exhibit quasi-periodic behaviors. Hamiltonian neural networks can maintain this property with a high probability even when the loss function does not completely vanish provided that the generalization error is small enough.

5 Behaviors of Hamiltonian neural networks for integrable Hamiltonian systems

5.1 Integrable Hamiltonian systems and a brief introduction to the KAM theory

In this section, we investigate the behaviors of the trained model for integrable Hamiltonian systems, particularly in the case where the loss function does not vanish completely. The Hamilton equation is the equation of the form (1) with a nondegenerate skew-symmetric matrix S:

$$\frac{\partial u}{\partial t} = S\frac{\partial H}{\partial u},\ S^\top = -S,\ \det S \neq 0.$$

For the matrix S to be nondegenerate, the dimension of the phase space must be an even number. Therefore we suppose that $u : t \in \mathbb{R} \mapsto u(t) \in \mathbb{R}^N,\ N = 2M$. As explained below, integrable systems exhibit quasi-periodic motions. The model (4) for Hamiltonian equations

$$\frac{\partial u}{\partial t} = S\frac{\partial H_{\mathrm{NN}}}{\partial u}$$

is called a Hamiltonian neural network. In Section 3, we have shown that the above model has the universal approximation property and hence the loss function can be arbitrarily small; however, although the model has the universal approximation property, in reality, minimization cannot be perfect, and a certain amount of error should be introduced.

In this section, we show that if the activation function of the neural network is sufficiently smooth and the loss function and hence the generalization error are not zero but sufficiently small, the trained model maintains with a high probability the above quasi-periodic behaviors of integrable systems. This result is obtained by the KAM theory and the generalization error analysis using the Rademacher complexities.

First, we briefly introduce some properties of Hamiltonian systems. Firstly, it is known as the Darboux theorem that by an appropriate coordinate transformation the matrix S is transformed into the normal form

$$\begin{pmatrix} O & I \\ -I & O \end{pmatrix}.$$

Hence it can be assumed that the matrix S is given as this form. Secondly, the following function $\omega : (v, w) \in \mathbb{R}^N \times \mathbb{R}^N \mapsto w(v, w) \in \mathbb{R}$

$$\omega(v, w) = v^\top S^{-1} w$$

is called the symplectic form. By using the symplectic form, a vector field X_F is associated to each function $F : \mathbb{R}^N \to \mathbb{R}$ by requiring

$$\omega(X_F, v) = \frac{\partial F}{\partial u} \cdot v \qquad \text{for all } v.$$

For two functions F, G, the following operation

$$\{F, G\} = \omega(X_F, X_G) \tag{8}$$

is called the Poisson bracket.

Definition 1. *A Hamiltonian system of which state variable is $N = 2M$ dimensional is integrable in the sense of Liouville if this Hamiltonian system has first integrals (i.e., conserved quantities) $F_1, F_2, \ldots, F_M$ with $\nabla F_1(u), \nabla F_2(u), \ldots, \nabla F_M(u)$ independent at each u and for all i, j*

$$\{F_i, F_j\} = 0.$$

For integrable systems, the following theorem is known.

Theorem 5 (Liouville–Arnold)**.** *Suppose that for an integrable Hamiltonian system there exist constants $c_1, \ldots, c_M$ such that $K = \cap_{i=1}^{M} F_i^{-1}(c_i)$ is connected and compact. Then there exist a neighborhood $\mathcal{N}$ of K and a coordinate transform*

$$\phi : (\theta, I) \in \mathbb{T}^n \times \mathcal{U} \to \phi(\theta, I) \in \mathcal{N} \tag{9}$$

such that the transformed system is the Hamilton equation of which Hamiltonian $H \circ \phi$ depends only on I.

The variables I and θ are called the action-angle variables. This theorem with the Darboux theorem roughly means that integrable Hamiltonian systems can be written in the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\theta \\ I\end{pmatrix} = \begin{pmatrix}O & I \\ -I & O\end{pmatrix}\begin{pmatrix}\frac{\partial \tilde{H}}{\partial \theta} \\ \frac{\partial \tilde{H}}{\partial I}\end{pmatrix}.$$

Further, since $\tilde{H} = H \circ \phi$ depends on I only, it holds that

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\theta \\ I\end{pmatrix} = \begin{pmatrix}O & I \\ -I & O\end{pmatrix}\begin{pmatrix}0 \\ \frac{\partial \tilde{H}}{\partial I}\end{pmatrix} = \begin{pmatrix}\frac{\partial \tilde{H}}{\partial I} \\ 0\end{pmatrix}.$$

This shows that I is constant and hence θ moves on the torus at a constant velocity. Because the velocity is typically not co-related with each other and hence the dynamics is "quasi-periodic." See, e.g., Scott Dumas (2014) for more details.

As seen above, integrable Hamiltonian systems are quasi-periodic. Therefore, the same property is preferably maintained when modeling such a system by neural networks. When the modelling error is sufficiently small, this is considered as a perturbation problem. The perturbation theory of Hamiltonian systems has been investigated from various aspects. For example, perturbed integrable Hamiltonian systems are in general no longer integrable; hence approximation of integrable Hamiltonian systems by integrable neural network models should be difficult. Fortunately, however, the KAM theory shows that even though the perturbed system is not integrable but maintains the above quasi-periodic behaviors under certain conditions.

The KAM theorem has many variants under various conditions. The following is a typical one (Scott Dumas, 2014).

Theorem 6 (KAM Theorem)**.** *Let θ, I be the action-angle variables for a C^∞ integrable Hamiltonian $H_0 : \mathbb{R}^{2M} \to \mathbb{R}$ with $M \geq 2$. If H_0 is non-degenerate, that is,*

$$\det \frac{\partial^2 H_0}{\partial I^2} \neq 0, \tag{10}$$

for the perturbed system

$$H(\theta, I) = H_0(I) + \varepsilon F(\theta, I, \varepsilon)$$

by $F \in C^\infty$, there exists ε_0 such that if $\varepsilon F < \varepsilon_0$, there exists a set of M-dimensional tori, invariant under the perturbed flow. On each invariant torus, the flow of the perturbed system H is quasiperiodic. In addition, the set of invariant tori is large in the sense that its measure becomes full as $\varepsilon \to 0$.

5.2 Existence of perturbed tori for the Hamiltonian neural networks

Suppose that an integrable Hamiltonian system with a compact phase space is modeled by using Hamiltonian neural networks. The universal approximation property shown in Section 3 guarantees that the value of MSE can be arbitrarily small; however, in actual training, a finite error remains. In this section, we apply the KAM theory to investigate the trained model in such cases.

To this end, we assume that the conditions for the Liouville–Arnold theorem, that is, for the target integrable system, there exist $c_1, \ldots, c_M$ such that $K = \cap_{i=1}^M F_i^{-1}(c_i)$ is connected and compact. We also assume that the Hamiltonian H is nondegenerate, which is the condition for the KAM theorem. Besides, since for application of the KAM theory the Hamiltonian and the perturbation must be smooth enough, we assume the smoothness of the activation functions of the neural networks.

Essentially, what we need to show is that the Hamiltonian H of the target system and H_{NN} of the trained model are sufficiently close to each other. Because in the learning process of Hamiltonian neural networks, the error on the energy gradient is typically minimized we need to develop a generalization error bound on the function values from the training errors of the derivatives.

The gap between the errors on the function values and those on the derivatives can be compensated by applying the Poincaré inequality (e.g., Chen & Hou (2020).) The Poincaré inequality essentially states that in order for a function to be large, its derivative must be large.

Theorem 7 (the Poincaré inequality)**.** *Suppose that $1 \leq p \leq \infty$ and $\Omega \subset \mathbb{R}^{2M}$ is bounded. Then there exists a constant c_{p} such that for any $H \in S_p^1(\Omega)$*

$$\int_\Omega |H(u) - \bar{H}|^p \mathrm{d}u \leq c_{\mathrm{p}} \left\| \frac{\partial H}{\partial u} \right\|_p^p,$$

where

$$\bar{H} = \frac{1}{\int_\Omega \mathrm{d}u} \int_\Omega H(u) \mathrm{d}u.$$

The constant c_{p} is called the Poincaré constant.

We use this inequality along with the invariance of the Hamilton equation under the constant shift of the energy function. More precisely, among the energy functions that yield the target Hamilton equation, we choose the one for which

$$\int H(u) \mathrm{d}u = \int H_{\mathrm{NN}}(u) \mathrm{d}u. \tag{11}$$

holds so that the error function has the zero mean:

$$e(u) := H(u) - H_{\mathrm{NN}}(u),\ \bar{e}(u) := 0.$$

Then from the Poincaré inequality we get

$$\int_\Omega |e(u)|^p \mathrm{d}u \leq c_{\mathrm{p}} \left\| \frac{\partial e}{\partial u} \right\|_{L^p}^p.$$

In addition, to apply the KAM theory, the error function $e(u)$ must be small not only for the data points but also for any point near the quasi-periodic solutions to the unperturbed system. Therefore, we need a generalization error bound for the function derivatives. In the following, we derive such an estimate.

Generalization error bounds are typically obtained by using the Rademacher complexities. See, e.g., Bousquet et al. (2004); Steinwart & Christmann (2008); Giné & Nickl (2016) for details.

Definition 2. *For a set of functions $\mathcal{G} \subset \{f \mid f : \mathcal{Z} \to \mathbb{R}\}$, the empirical Rademacher complexity of $\mathcal{G}$ with respect to $z \in \mathcal{Z}^n$ is defined by*

$$\hat{\mathcal{R}}(\mathcal{G}) := E_\sigma \left[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n \sigma_i g(z_i) \right],$$

where E_σ is the expectation with respect to the uniform distribution of $\sigma \in \{-1, 1\}^n$. In addition, if the data $z = (z_1, z_2, \ldots, z_n)$ are distributed according to a probability measure P, then the Rademacher complexity of $\mathcal{G}$ with respect to P is defined by

$$\mathcal{R}_n(\mathcal{G}) := E_P \left[\hat{\mathcal{R}}(\mathcal{G}) \right]$$

The following is a typical generalization bound using the Rademacher complexity.

Theorem 8. *Let X, Y be arbitrary spaces, $\mathcal{F} \subset \{f : X \to Y\}$ be a hypotheses class and $L : Y \times Y \to [0, c]$ be a loss function. Let $\mathcal{G}$ be defined by $\{(x, y) \in X \times Y \mapsto L(y, h(x)) \in \mathbb{R} \mid h \in \mathcal{F}\}$. Then for any $\delta > 0$ and any probability measure P, we have with a probability at least $(1 - \delta)$ with respect to the repeated sampling of P^n-distributed training data*

$$E[L(Y, h(X))] - \frac{1}{n}\sum_{i=1}^{n} L(Y_i, h(X_i)) \leq 2\mathcal{R}_n(\mathcal{G}) + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$$

for all $h \in \mathcal{F}$.

Although we only need the generalization error bound for the first-order derivatives, in what follows, we show a more general bound. Suppose that the model is trained by a certain kind of the Sobolev training (Czarnecki et al., 2017) where the errors measured by

$$\sum_{j=1}^{l} \left\| \frac{\partial H(u)^j}{\partial u^j} - \frac{\partial H_{\mathrm{NN}}(u)^j}{\partial u^j} \right\|_p^p \tag{12}$$

is minimized. We denote the number of the components of the derivatives that appear in the above summation by N'. By applying Theorem 8 to the approximation of N'-dimensional vector-valued functions, we obtain

Lemma 2. *Let X be the phase space K and Y be*

$$Y = \{y = (y_1, y_2, \ldots, y_{N'}) \in \mathbb{R}^{N'} \mid {}^{\exists}H, y_1 = \frac{\partial H}{\partial u_1}, y_2 = \frac{\partial H}{\partial u_2}, \ldots\}.$$

$\mathcal{F} \subset \{f : X \to Y\}$ be a hypotheses class and $L : Y \times Y \to [0, c]$ be the loss function

$$L(Y, Y') = \sum_{i=1}^{N'} \|Y_i - Y_i'\|_p^p$$

Let $\mathcal{G}$ be defined by $\{(x, y) \in X \times Y \mapsto L(y, h(x)) \in \mathbb{R} \mid h \in \mathcal{F}\}$. Then for any $\delta > 0$ and any probability measure P, we have with a probability at least $(1 - \delta)$ with respect to the repeated sampling of P^n-distributed training data

$$E[L(Y, h(X))] - \frac{1}{n}\sum_{i=1}^{n} L(Y_i, h(X_i)) \leq 2\mathcal{R}_n(\mathcal{G}) + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$$

for all $h \in \mathcal{F}$.

Note that the terms in the above estimate corresponds to

$$E[L(Y, h(X))] = \int \sum_{j=1}^{l} \left\| \frac{\partial H(u)^j}{\partial u^j} - \frac{\partial H_{\mathrm{NN}}(u)^j}{\partial u^j} \right\|_p^p \mathrm{d}P,$$

$$\frac{1}{n}\sum_{i=1}^{n} L(Y_i, h(X_i)) = \sum_{j=1}^{N_{\mathrm{data}}} \int \sum_{j=1}^{l} \left\| \frac{\partial H(u)^j}{\partial u^j}(u_{\mathrm{d},j}) - \frac{\partial H_{\mathrm{NN}}(u)^j}{\partial u^j}(u_{\mathrm{d},j}) \right\|_p^p$$

respectively, where N_{data} is the number of the data points and $u_{\mathrm{d},j}$ is the each data point. Then, if we relax the space Y to $\tilde{Y} = \mathbb{R}^{N'}$, corresponding function space

$$\tilde{\mathcal{G}} = \{(x, y) \in X \times \tilde{Y} \mapsto L(y, h(x)) \in \mathbb{R} \mid h \in \mathcal{F}\}.$$

contains $\mathcal{G}$. From the property of the Rademacher complexities $\mathcal{G} \subset \tilde{\mathcal{G}}$ implies $\mathcal{R}_n(\mathcal{G}) \leq \mathcal{R}_n(\tilde{\mathcal{G}})$. Thus we have

Theorem 9. *Under the same condition as Lemma 2, for any $\delta > 0$ and any probability measure P, we have with a probability at least $(1 - \delta)$ with respect to the repeated sampling of P^n-distributed training data*

$$E[L(Y, h(X))] - \frac{1}{n}\sum_{i=1}^{n} L(Y_i, h(X_i)) \leq 2\mathcal{R}_n(\tilde{\mathcal{G}}) + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$$

for all $h \in \mathcal{F}$.

Fortunately, $\mathcal{R}_n(\tilde{\mathcal{G}})$ can be bounded. In fact, the Rademacher complexities of the neural networks are well studied and there exist theoretical bounds depending on the architecture of the networks (Neyshabur et al., 2015; Golowich et al., 2018). In addition, because the phase space is assumed to be compact the loss function is Lipschitz. Because the Rademacher complexities of the composition of a function and a Lipschitz function is estimated by the Rademacher complexities of the former function multiplied by the Lipschitz constant of the latter function, $\mathcal{R}_n(\tilde{\mathcal{G}})$ is bounded by a constant proportional to the Rademacher complexity of the employed neural networks.

From the above estimate we have

$$\int \sum_{j=1}^{l} \left\| \frac{\partial H(u)^j}{\partial u^j} - \frac{\partial H_{\mathrm{NN}}(u)^j}{\partial u^j} \right\|_p^p \mathrm{d}P \leq \frac{1}{n} \sum_{i=1}^{n} L(Y_i, h(X_i)) + 2\mathcal{R}_n(\tilde{\mathcal{G}}) + c\sqrt{\frac{\ln \frac{1}{\delta}}{2n}}$$

In addition, if we assume that there exists a density f_P for the measure P with $\inf f_P > 0$, the above inequality gives a bound on

$$\int \sum_{j=1}^{l} \left\| \frac{\partial H(u)^j}{\partial u^j} - \frac{\partial H_{\mathrm{NN}}(u)^j}{\partial u^j} \right\|_p^p \mathrm{d}u.$$

which is essentially the Sobolev norm in $W^{p,l}$ (Adams & Fournier, 2003). Because in general the Rademacher complexity is a decreasing function of n, the right hand side other than the loss function of the training process decreases as n becomes large. Hence if the training loss is small enough and also the number of data is large enough, the norm of the error function e in $W^{p,l}$ can be small with a probability at least $(1-\delta)$. Lastly, we use the Sobolev inequality (Adams & Fournier, 2003; Benyi & Oh, 2013); there exist constants c_1, c_2 such that if $lp > d$

$$\|e\|_{L^\infty}(\mathbb{R}^d) \leq c\|e\|_{W^{p,l}}(\mathbb{R}^d), \quad \|e\|_{L^\infty}(\mathcal{T}^d) \leq c\|e\|_{W^{p,s}}(\mathcal{T}^d)$$

to ensure that H and H_{NN} is close in the function values. In particular, we consider the case $l = 1$ because typically the loss function (12) with $l = 1$ is used in Hamiltonian neural networks. Thus, to apply the above Sobolev inequality we need $p > d$.

By combining the above results, we have the following theorem.

Theorem 10. *Suppose that the target system is an integrable Hamiltonian system with a C^∞ and non-degenerate Hamiltonian $H : \mathbb{R}^{2M} \to \mathbb{R}$ with $M \geq 2$. Suppose also that a Hamiltonian neural network with C^∞ activation functions is used to model the target system. Then for a fixed δ, if the loss function*

$$0 < \frac{1}{N_{\mathrm{data}}} \left\| \frac{\partial H}{\partial u} - \frac{\partial H_{\mathrm{NN}}}{\partial u} \right\|_p$$

for $p > 2M$ is small enough, the number of the data N_{data} is large enough so that the generalization error and hence the sup norm of $\|e\|_{L^\infty}$ is less than a constant ε_0, there exists a set of invariant tori for the trained model H_{NN} with a probability at least $(1-\delta)$. The constant ε_0 depends on δ, the Poincaré constant c_{p}, $\inf f_P$ and the threshold of the KAM theorem.

6 Concluding Remarks

In this paper, we have investigated the approximation properties of the deep energy-based models, including Hamiltonian neural networks. In particular, we have provided the universal approximation theorem for the general deep energy-based models with and without the latent variables. In addition, the general energy-based model with the latent variables is newly introduced in this paper. For the integrable Hamiltonian systems, we have applied the KAM theory to prove the persistence of the quasi-periodic behaviors with a high probability even when the loss function is not perfectly zero. Future work includes a more sophisticated analysis of the generalization errors and similar analysis for the other types of neural networks for physics, such as the variational integrator networks.

References

Adams, R. A. and Fournier, J. J. F. *Sobolev Spaces*. Elsevier, June 2003.

Benyi, A. and Oh, T. The Sobolev inequality on the torus revisited. *Publ. Math. Debrecen*, 83(3):359–374, October 2013.

Bharadwaj, P., Li, M., and Demanet, L. SymAE: An autoencoder with embedded physical symmetries for passive time-lapse monitoring. In *SEG Technical Program Expanded Abstracts 2020*. Society of Exploration Geophysicists, 2020.

Bondesan, R. and Lamacraft, A. Learning Symmetries of Classical Integrable Systems. In *ICML 2019 Workshop on Theoretical Physics for Deep Learning*, 2019.

Bousquet, O., Boucheron, S., and Lugosi, G. Introduction to statistical learning theory. In Bousquet, O., von Luxburg, U., and Rätsch, G. (eds.), *Advanced Lectures on Machine Learning*, pp. 169–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

Caginalp, G. An analysis of a phase field model of a free boundary. *Archive for Rational Mechanics and Analysis*, 92 (3):205–245, 1986.

Chen, T. Q., Rubanova, Y., Bettencourt, J., Duvenaud, D., Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Chen, Y. and Hou, T. Y. Function Approximation via the Subsampled Poincaré Inequality. *Discrete & Continuous Dynamical Systems - A*, 41(1):169–199, August 2020.

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. Symplectic Recurrent Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian Neural Networks. *ICLR 2020 Deep Differential Equations Workshop*, 2020.

Czarnecki, W. M., Osindero, S., Jaderberg, M., Swirszcz, G., and Pascanu, R. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4281–4290, 2017.

Desai, S. and Roberts, S. VIGN: Variational Integrator Graph Networks. *arXiv*, 2020.

DiPietro, D. M., Xiong, S., and Zhu, B. Sparse Symplectically Integrated Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Feng, Y., Wang, H., Yang, H., and Wang, F. Time-Continuous Energy-Conservation Neural Network for Structural Dynamics Analysis. *arXiv*, 2020.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data. In *International Conference on Machine Learning (ICML)*, pp. 3165–3176, 2020a.

Finzi, M., Wang, K. A., and Wilson, A. G. Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Furihata, D. and Matsuo, T. *Discrete Variational Derivative Method: A Structure-Preserving Numerical Method for Partial Differential Equations*. Chapman and Hall/CRC, 2010.

Galioto, N. and Gorodetsky, A. A. Bayesian Identification of Hamiltonian Dynamics from Symplectic Data. In *IEEE Conference on Decision and Control (CDC)*, pp. 1190–1195, 2020.

Giné, E. and Nickl, R. *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge University Press, 2016.

Golowich, N., Rakhlin, A., and Shamir, O. Size-Independent sample complexity of neural networks. In *Conference On Learning Theory (COLT)*, pp. 297–299, 2018.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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., Kevrekidis, I. G., and Karniadakis, G. E. Learning Poisson systems and trajectories of autonomous systems via Poisson neural networks. *arXiv*, 2020a.

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, 2020b.

Jin, Z., Lin, J. Y.-Y., and Li, S.-F. Learning principle of least action with reinforcement learning. *arXiv*, 2020c.

Langley, P. Crafting Papers on Machine Learning. In *International Conference on Machine Learning (ICML)*, pp. 1207–1216, 2000.

Li, S.-H., Dong, C.-X., Zhang, L., and Wang, L. Neural Canonical Transformation with Symplectic Flows. *Physical Review X*, 10(2):021020, 2020.

Marsden, J. E. and Ratiu, T. S. *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*. Springer Science & Business Media, March 2013.

Matsubara, T., Ishikawa, A., and Yaguchi, T. Deep Energy-Based Modeling of Discrete-Time Physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Miehe, C., Hofacker, M., and Welschinger, F. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering*, 199(45):2765–2778, 2010.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-Based capacity control in neural networks. In *Conference on Learning Theory (COLT)*, pp. 1376–1401, 2015.

Sæmundsson, S., Hofmann, K., Terenin, A., and Deisenroth, M. P. Variational integrator networks for physically meaningful embeddings. In *Artificial Intelligence and Statistics (AISTATS)*, volume 108, pp. 3078–3087, 2019.

Sanchez-Gonzalez, A., Bapst, V., Cranmer, K., and Battaglia, P. Hamiltonian Graph Networks with ODE Integrators. *arXiv*, 2019.

Scott Dumas, H. *KAM Story, The: A Friendly Introduction To The Content, History, And Significance Of Classical Kolmogorov-Arnold-Moser Theory*. World Scientific Publishing Company, 2014.

Steinbach, I. Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001, 2009.

Steinwart, I. and Christmann, A. *Support Vector Machines*. Springer Science & Business Media, September 2008.

Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. Hamiltonian Generative Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Wheeler, A. A., Murray, B. T., and Schaefer, R. J. Computation of dendrites using a phase field model. *Physica D*, 66 (1):243–262, 1993.

Xiong, S., Tong, Y., He, X., Yang, C., Yang, S., and Zhu, B. Nonseparable Symplectic Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control. In *International Conference on Learning Representations (ICLR)*, 2020.

A Numerical results of the energy-based physical model with latent variables

Although the main contribution of this paper is the theoretical analyses of the deep energy-based physical models, to the best of our knowledge, the model with latent variables

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (\frac{\partial u}{\partial x})^{-1} G (\frac{\partial u}{\partial x})^{-\top} \frac{\partial H}{\partial x}.$$

shown in Section 4 in the main paper is new. For a better understanding of the significance of the theoretical results presented in this paper, we performed numerical tests using this model. However, we would like to emphasize that the proposal of this model is not the main purpose of this study.

Firstly, we consider a double pendulum shown in Figure 3, of which equation of motion is

$$\begin{aligned}
&\frac{\mathrm{d}\theta_1}{\mathrm{d}t} = \phi_1, \quad \frac{\mathrm{d}\theta_2}{\mathrm{d}t} = \phi_2, \\
&\frac{\mathrm{d}\phi_1}{\mathrm{d}t} = \frac{g(\sin\theta_2 \sin(\theta_1-\theta_2) - \frac{m_1+m_2}{m_2}\sin(\theta_1)) - (l_1\theta_1^2\cos(\theta_1-\theta_2) + l_2\theta_2^2)\sin(\theta_1-\theta_2)}{l_1(\frac{m_1+m_2}{m_2} - \cos^2(\theta_1-\theta_2))}, \\
&\frac{\mathrm{d}\phi_2}{\mathrm{d}t} = \frac{\frac{g(m_1+m_2)}{m_2}(\sin\theta_1\cos(\theta_1-\theta_2) - \sin(\theta_2)) - (\frac{l_1(m_1+m_2)}{m_2}\theta_1^2 + l_2\theta_2^2\cos(\theta_1-\theta_2))\sin(\theta_1-\theta_2)}{l_2(\frac{m_1+m_2}{m_2} - \cos^2(\theta_1-\theta_2))}.
\end{aligned} \tag{13}$$

The energy function of this system is

$$H = \frac{1}{2}(m_1+m_2)l_1^2\phi_1^2 + \frac{1}{2}m_2l_2^2\phi_2^2 + m_2l_1l_2\phi_1\phi_2\cos(\theta_1-\theta_2) + g(m_1+m_2)l_1\cos\theta_1 + gm_2l_2\cos(\theta_2)$$

and the Lagrangian is

$$\mathcal{L} = \frac{1}{2}(m_1 + m_2)l_1^2\phi_1^2 + \frac{1}{2}m_2 l_2^2\phi_2^2 + m_2 l_1 l_2 \phi_1 \phi_2 \cos(\theta_1 - \theta_2) - g(m_1 + m_2)l_1 \cos\theta_1 - g m_2 l_2 \cos(\theta_2).$$

Note that the generalized momenta of this system are not obvious:

$$p_1 = \frac{\partial \mathcal{L}}{\partial \phi_1} = (m_1 + m_2)l_1^2\phi_1 + m_2 l_1 l_2 \phi_2 \cos(\theta_1 - \theta_2),$$
$$p_2 = \frac{\partial \mathcal{L}}{\partial \phi_2} = m_2 l_2^2 \phi_2 + m_2 l_1 l_2 \phi_1 \cos(\theta_1 - \theta_2).$$

Because these are difficult to be observed, we suppose that the values of the state variables θ_1, θ_2 and their derivatives ϕ_1, ϕ_2 are given as data. In such a situation, Hamiltonian neural networks (Greydanus et al., 2019)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ q_2 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} O & I \\ -I & O \end{pmatrix} \nabla H_{\mathrm{NN}}(q_1, q_2, p_1, p_2)$$

are not appropriate because this model can be used only when p_1 and p_2 are generalized momenta; in the case considered here, they are supposed to be unknown.

To confirm it, we tested a naive model

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} O & I \\ -I & O \end{pmatrix} \nabla H_{\mathrm{NN}}(q_1, q_2, v_1, v_2), \tag{14}$$

where v_1, v_2 are not the generalized momenta but the velocities $v_1 = \dot{q}_1, v_2 = \dot{q}_2$, and the proposed model

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (\frac{\partial u}{\partial x})^{-1} G (\frac{\partial u}{\partial x})^{-\top} \frac{\partial H_{\mathrm{NN}}}{\partial x}.$$

In the experiments, the programs are implemented using Python 3.8.5 with the packages PyTorch 1.7.1, numpy 1.19.5, scipy 1.6.0, autograd 1.3, torchdiffeq 0.2.1. As the data, we used numerical solutions to (13) with the parameters $l_1 = l_2 = 1.0$, $m_1 = 1$, $m_2 = 2$, $g = 9.8$ solved by the scipy odeint with 100 initial conditions that are randomly generated from the standard normal distribution. We numerically integrated each orbit on the time interval $[0, 5]$, in which the computed values are evaluated at 100 points with the identical sampling rate. Then the target data

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \theta_1 \\ \theta_2 \\ \phi_1 \\ \phi_2 \end{pmatrix}$$

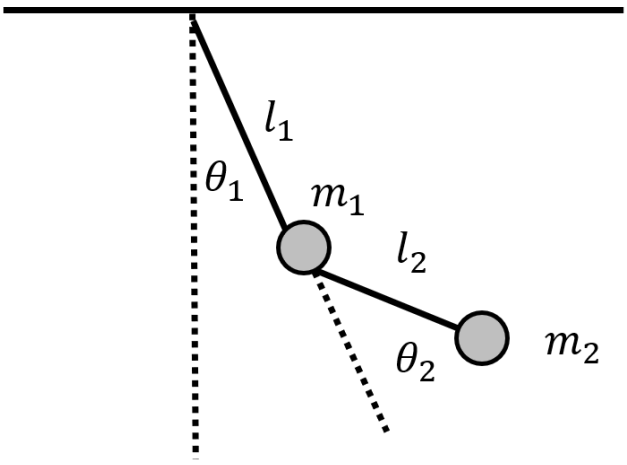
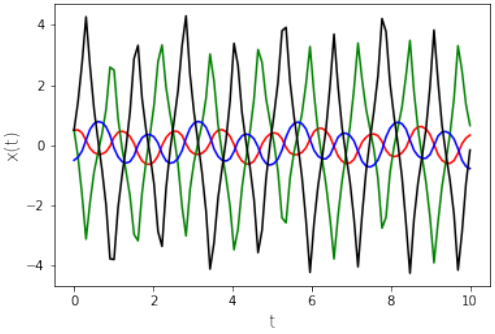
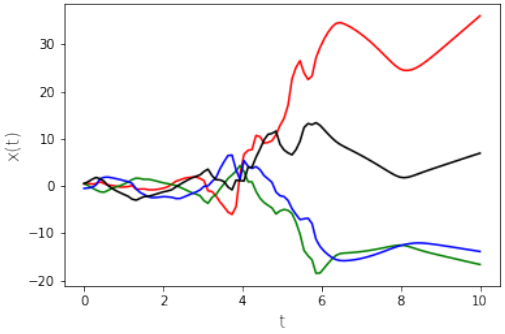


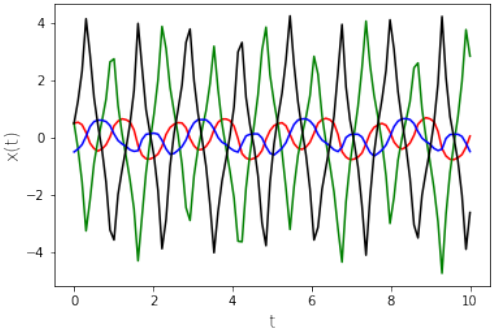
Figure 3: The double pendulum used as the target in the first experiment.



(a) Ground truth



(b) Naive model



(c) Proposed model

Figure 4: An example of the predicted orbits by the Hamiltonian neural network and the proposed model. Each component of $x(t) = (q_1(t), v_1(t), p_2(t), v_2(t)$ is represented by the red (q_1), green (v_1), blue (q_2), and black (v_2) curves.

are obtained by substituting the computed state variables into the right-hand side of (13). The experiments were performed on GeForce 2080 Ti. We used the Adam algorithm with the learning rate 0.001. The number of the learning steps was 10000 and the batch size was set to 200. In both cases, the energy function is modeled by using a multilayer perceptron with $\tanh$ as the activation function. The network has only one hidden layer, of which size is set to 50.

Remark 3. *For the coordinate transformations in the proposed model, the invertible neural networks should be used; however we used the simple multilayer perceptron here because this model is assumed in the theorem in the main part of the paper. In the theorem, we needed neural networks that can approximate given functions and their derivatives. Although the universal approximation property of invertible neural networks is proved recently (Teshima et al., 2020), this theorem shows the universal approximation property of the function values only, not of the derivatives, which are needed for computation of the Jacobi matrix in our model.*

The examples of the predicted orbits are shown in Figure 4. The training losses of these models were 13.6 for the Hamiltonian neural network and 0.280 for the proposed model. As shown in this figure, the naive model failed to capture the dynamics correctly. This is because the dynamics of $\theta_1, \theta_2, \phi_1, \phi_2$ cannot be described by (14). This result illustrates that in order to model the physical phenomena by using the model of the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = G\frac{\partial H}{\partial x},$$

the choice of the coordinate system is important.

Mass-spring system. Due to the well-known chaotic behaviors of the double pendulum, the results, in particular the values of the losses, of the previous experiments are to a certain extent unstable, except for the fact that Hamilton neural networks always failed. Therefore, secondly, we investigated the models in more detail using a simple mass-spring system depicted in Figure 5. The two mass points m_1 and m_2 are connected by the springs, each of which respectively

has the spring constant k_1 and k_2 and the natural length l_1 and l_2. This system is a Hamiltonian system with the energy function

$$H(q_1, q_2, p_1, p_2) = \frac{p_1^2}{2m_1} + \frac{p_2^2}{2m_2} + \frac{k_1(q_1 - l_1)^2}{2} + \frac{k_2(q_2 - q_1 - l_2)^2}{2}, \tag{15}$$

where q_1, q_2 are the positions of the mass points and p_1, p_2 are the momenta, which are defined by $p_1 = m_1 v_1, p_2 = m_2 v_2, v_1 = \mathrm{d}p_1/\mathrm{d}t, v_2 = \mathrm{d}p_2/\mathrm{d}t$. Suppose that we do not know the exact values of m_1 and m_2 and the only positions q_1 and q_2 and their derivatives can be observed. Although m_1 and m_2 may be estimated from the data, for evaluation of the models, we tried to model the dynamics only using q_1, q_2 and their derivatives.

We examined the naive model, the proposed model and the neural ordinary differential equation (Chen et al., 2018)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{pmatrix} = f_{\mathrm{NN}}(q_1, q_2, v_1, v_2).$$

The condition of the experiment is almost the same as the previous one, except for the batch size, which we set to 100 in this experiment. As the data, we used numerical solutions to

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ -\frac{k_1}{m_1}(q_1 - l_1) + \frac{k_2}{m_1}(q_2 - q_1 - l_2) \\ -\frac{k_2}{m_2}(q_2 - q_1 - l_2) \end{pmatrix}, \tag{16}$$

which is equivalent to (15).

Examples of the predicted orbits are shown in Figure 6. While the proposed method gave almost exactly the same orbit as the ground truth, the naive model failed to predict the states. In fact, it should be impossible to rewrite the equation (16) to (14) with a certain energy function; for example, when the system has just one mass point and the equation of motion is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} v_1 \\ -\frac{k_1}{m_1}(q_1 - l_2) \end{pmatrix},$$

this can be transformed into a Hamiltonian system

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} q_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \nabla \tilde{H},\ \tilde{H} = \frac{v_1^2}{2m_1'} + k_1'(q_1 - l_1)^2$$

with $k_1' = k_1/m_1, m_1'$ and the mass m_1' is 1. Hence, for this system, the Hamiltonian neural networks are applicable without the knowledge of m_1. However, for the system (16) such a transformation cannot be applied. Meanwhile, in the proposed model, a coordinate transformation that makes the equation Hamiltonian is explored.

The result by the neural ODE is better than that of the naive model; however, the prediction error is increased as time becomes large. This is due to the non-existence of the energy conservation law for the neural ODE. In fact, the value of the energy function of the neural ODE model was increasing, as shown in Figure 7.

The losses for the above models are shown in Table 1. We performed twelve experiments for each model and the losses for these experiments are listed. The values of the loss functions of the neural ODE are very small and those of the

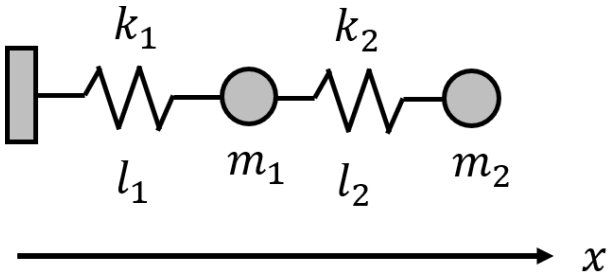
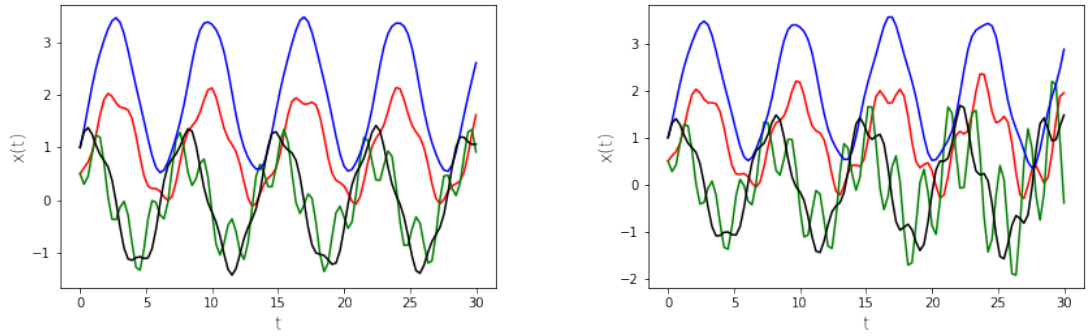
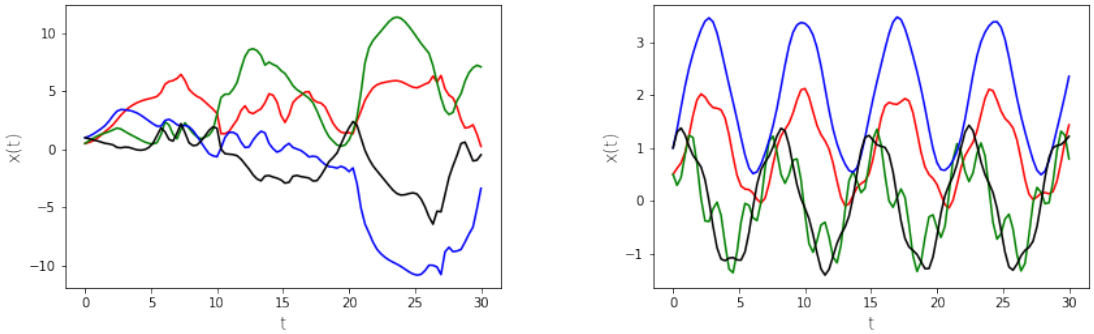


Figure 5: The target mass-spring system used in the second experiment.

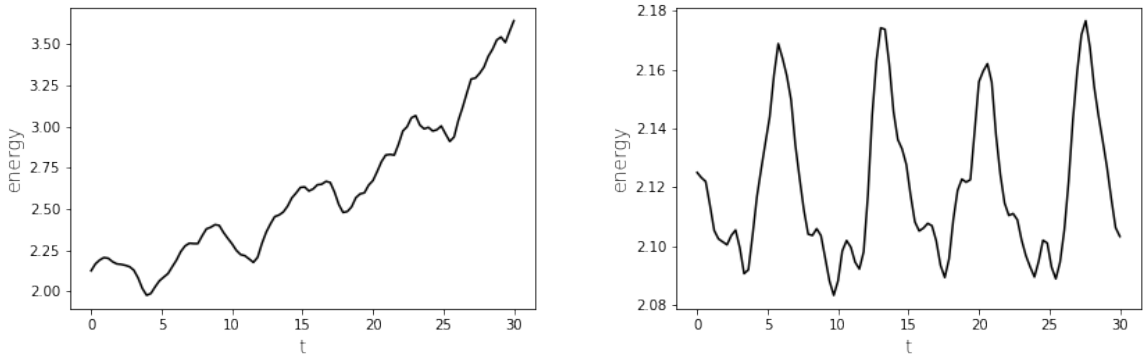


(a) Ground truth (b) NODE



(c) Naive model (d) Proposed model

Figure 6: An example of the predicted orbits by the neural ODEs (NODE), Hamiltonian neural networks and the proposed model. Each component of $x(t) = (q_1(t), v_1(t), p_2(t), v_2(t)$ is represented by the red (q_1), green (v_1), blue (q_2), and black (v_2) curves.



(a) NODE (b) Proposed model

Figure 7: The values of the energy function predicted by the neural ODE model and by the proposed model.

proposed model were not stable. Actually, the histories of the loss functions for the proposed model were not monotonic at all. This is due to the non-uniqueness of the proposed model; if a model is fit to the given data quite well, then so are its canonical transformations. Due to this feature of the model, after the loss function went down, it sometimes went up significantly in the search for different coordinate transformations. If the learning process is successful, the values of the loss function are competitive to the neural ODEs; however, the neural ODEs exhibit the energy drift as seen in the above and hence not suitable for long-term predictions.

Table 1: The training and test losses of the neural ordinary differential equations (NODE), the naive model and the proposed model (LHNN).

NODE		HNN		LHNN	
train	test	train	test	train	test
0.000361	0.00223	0.566	15.6	0.00697	0.00823
0.000221	0.00101	0.614	18.1	17.9	3214.7
0.0000209	0.000605	0.630	17.0	0.00198	0.00688
0.000197	0.000612	0.527	17.4	0.000253	0.00254
0.000128	0.000552	0.637	15.9	24.2	3247.7
0.000258	0.00104	0.603	17.3	0.000954	0.0101
0.000999	0.00150	0.586	16.2	0.000550	0.00996
0.000527	0.00181	0.583	18.9	0.230	2.96
0.000307	0.00164	0.572	19.6	0.000565	0.00513
0.0000450	0.000742	0.599	18.2	0.00316	0.00722
0.0000522	0.00101	0.627	17.9	0.000615	0.00401
0.0000374	0.000931	0.614	16.1	0.00909	0.229