

GENERALIZATION ERROR ANALYSIS OF DEEP PHYSICAL MODELS WITH LATENT VARIABLES TRAINED ON TRAJECTORY DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

In this paper, we investigate the generalization error of deep physical models with latent variables. Deep physical models, such as Hamiltonian Neural Networks, are neural network models for learning equations of motion from observational data of physical phenomena and have attracted much attention in recent years. In particular, in such cases, the data are not completely random, but rather given as random trajectories. We provide an error bound for the case where the training data are given in such a way. Our results show that it is important to collect data from many trajectories, rather than simply collecting a large number of data, to improve generalization performance. In addition, an important application of the combination of deep physics models with latent variables is the interpolation of images from videos while preserving the laws of physics, such as the energy conservation law. However, when the frame interval of the video is large, it can be difficult to preserve the laws of physics. In this paper, we show that it is possible to interpolate the images from videos so that the laws of physics are preserved, provided that the generalization error is sufficiently small.

1 INTRODUCTION

Recently, deep learning methods have attracted much attention for learning the equations of motion from observational data of physical phenomena. In this paper, we investigate the generalization error of such models when the models are trained with the random trajectory data. In addition, an important application of deep physical models is the interpolation of images from videos while preserving the laws of physics, such as the energy conservation law. As an application of the error analysis, we also show that it is possible to interpolate the images from videos so that the laws of physics are preserved, provided that the generalization error is sufficiently small.

Although neural networks have been applied not only to modeling physical phenomena but also to other important tasks that include solving partial differential equations such as physics-informed neural networks (Raissi et al., 2019) and neural operators (Li et al., 2020; Kovachki et al., 2023; Lu et al., 2021), in this paper we focus on the neural network models for modeling and call them deep physics models for simplicity.

Typically behind the physical phenomena is analytical mechanics, which is the theory of classical mechanics that allows the modeling of complex and non-linear dynamical systems. There are two theories of analytical mechanics, the Lagrange and the Hamilton mechanics. Most of the deep physical models are based on the Hamilton equation, which is the equation of motion of Hamiltonian mechanics:

$$\frac{du}{dt} = S \frac{\partial H}{\partial u}, \tag{1}$$

where $u : t \in \mathbb{R} \mapsto u(t) \in \mathbb{R}^N$, $H : u \in \mathbb{R}^N \mapsto H(u) \in \mathbb{R}$. Typically, u is given as (q, p) , where q represents the state variables and p the generalized momenta. S is a skew-symmetric matrix and H is a real-valued function of u , which represents the total energy of the system. The most representative model for extracting the equation of motion is the Hamiltonian neural networks (HNNs, Greydanus

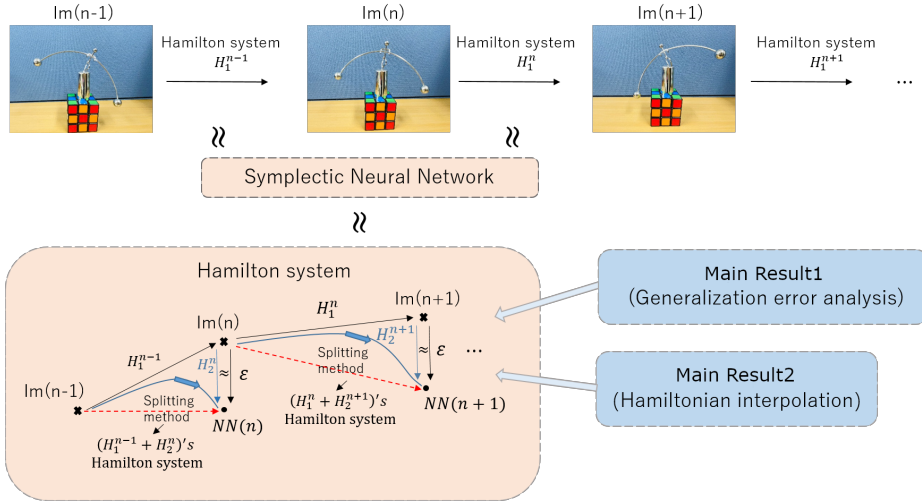


Figure 1: Overview of the research. Interpolation is performed on a sequence of images representing a physical phenomenon while respecting the laws of physics. If the modeling error of the learned model is small enough, the difference between the true Hamiltonian and the model can be approximated by a Hamiltonian system with very high accuracy. This can be used to construct a Hamiltonian system that approximates the model.

et al. (2019))

$$\frac{du}{dt} = S \frac{\partial H_{NN}}{\partial u}, \quad (2)$$

where H_{NN} is a neural network, and various extensions of this model have been proposed. Based on the theory of physics, these models are designed so that the laws of physics, such as the energy conservation law, are maintained.

On the other hand, theoretical analysis is limited and, in particular, no error analysis has been performed in the practical situations where the models are trained with trajectory data. A generalization error bound of HNNs is investigated in Chen et al. (2021a), in which the training data are assumed to be given completely at random. However, when these models are used, the data are not completely random, but rather given as random trajectories. In addition, a typical application of such models is the interpolation of the images from videos while preserving the laws of physics. For this application, the models are combined with the autoencoder; however, to the best of the authors’ knowledge, there are no theoretical analyses of the performance of such models. In particular, as explained in Section 3.2, it is not obvious whether the trained models in fact preserve the laws of physics or not.

In this study, first, we present a generalization error bound for the deep physical models including HNNs with and without the autoencoder. Theoretical analysis of generalization error is typically based on statistical learning theory, in which it is usually assumed that the data are random samples from a certain probability distribution, thereby evaluating the difference between the loss function for the training data and the expected value of the loss function. However, as explained above, when neural networks for modeling the equation of motion are used, the training data are given as trajectory data. More precisely, the initial values of the data are given randomly, but the subsequent data are generated according to the laws of physics. Therefore, only the initial value of each trajectory can be given at random; as far as we know, this problem setting has not been considered yet in the existing theoretical analyses for such models.

Second, as an application of the generalization error analysis, we prove that the so-called Hamiltonian interpolation is possible by combining the autoencoder and symplectic neural network models. As is demonstrated in the original paper of HNN (Greydanus et al., 2019), an important application of these models is learning the equation of motion from images by combining the autoencoder, thereby interpolating the images while preserving the laws of physics. For example, when creating an animation, if several images with large time intervals can be prepared and interpolated between them, the animation can be created efficiently. To avoid unnaturalness, it is desirable to interpolate

Table 1: Comparison with other theoretical results for deep physical models.

| | SympNet (Jin et al., 2022) | GFINNs (Zhang et al., 2022) | KAM (Chen et al., 2021a) | Our Results |
|---------------------------------|----------------------------------|-----------------------------------|--------------------------------|-------------|
| Universal Approximation Theorem | yes | yes | yes | |
| Generalization Error Analysis | | | yes | yes |
| Learning from Pixels | | | | yes |
| Random Trajectory Data | | | | yes |

the images while respecting the laws of physics. In such an application, whether the interpolated images follow a Hamiltonian equation is very important. Meanwhile, the problem of determining if a given function is obtained from a Hamiltonian equation or not has been studied in the fields of symplectic geometry and numerical analysis. By combining the generalization error bound and these theories, particularly the so-called splitting method, we will show that if the number of data is sufficient, with a high probability the interpolation is possible while preserving the laws of physics.

The main contributions of this paper are as follows (see also Figure 1 and Table 1.)

- We establish a theoretical framework for analyzing the generalization error when the initial values of each trajectory in the data can be given at random. As far as we know, there is no theoretical analysis performed in this setting. It is shown that the generalization error depends on the number of trajectories, not on the total number of the data.
- By combining this generalization error analysis with the theory of symplectic geometry and the error analysis of a numerical integrator called the splitting method, we theoretically prove that with a high probability, there exists a Hamiltonian system that interpolates the given image sequences with large time intervals. This result presents a new direction in the theoretical analysis of deep physical models in the sense that it is a combined study of three fields: statistical learning theory, symplectic geometry, and numerical analysis.

2 RELATED WORK

Since ordinary differential equations describe general dynamical systems, Neural Ordinary Differential Equation (NODE, Chen et al. (2018)) is a general model for modeling ordinary differential equations. However, due to the generality of neural networks, this method does not necessarily work well for modeling physical phenomena. Most of the existing deep learning models for physics are based on analytical mechanics. For example, the Lagrangian Neural Networks (LNN, Cranmer et al. (2020)) learns the Lagrangian from data and derives the equation of motion as the Euler–Lagrange equation. Hamiltonian Neural Networks (HNN, Greydanus et al. (2019)) is a neural network model for Hamiltonian systems, in which the Hamiltonian, the energy function, is modeled using neural networks. There are many extensions of HNNs, for example, symplectic ODE-Net (Zhong et al., 2019), Variational Integrator Networks (Saemundsson et al., 2019), DGNet(Matsubara et al., 2019), neural symplectic forms (Chen et al., 2021b), Poisson Neural Networks (Jin et al., 2022).

At the same time, these deep physical models are applied to derive equations of motion from animations. The currently available method includes the Hamiltonian Generative Network (Toth et al., 2019), which is capable of learning Hamiltonian dynamics from high-dimensional observations (e.g. images.) Similarly, KeyCLD(Daems et al., 2022) is a framework for learning Lagrangian dynamics from images. Khan & Storkey (2021) focused on a deep generative model utilizing Hamiltonian Latent Operators to reliably disentangle content and motion information in image sequences. Mason et al. (2022) presented a physics-informed neural network model to estimate and predict 3D rotational dynamics from image sequences with a multi-stage prediction pipeline. Hofherr et al. (2022) proposed to combine neural implicit representations for appearance modeling with neural ordinary differential equations for modeling physical phenomena to obtain a dynamic scene representation from visual observations. Jatavallabhula et al. (2021) proposed gradsim which addressed visuomotor control tasks without relying on state-based supervision.

From the perspective of theoretical research, Jin et al. (2020) proposed SympNet along with the universal approximation theorem for the proposed model. HNNs were shown to have the universal approximation property in Chen et al. (2021a) along with an application to the KAM theory. Zhang et al. (2022) proposed the GENERIC formalism informed neural networks that obey the degeneracy conditions of the GENERIC formalism, and proved the universal approximation theorem for their models.

As far as we know, no theoretical analysis of deep learning models trained with trajectory data has been performed. In addition, the Hamiltonian interpolation using these models has not been investigated.

3 BRIEF INTRODUCTION OF DEEP PHYSICAL MODELS WITH LATENT VARIABLES AND SUMMARY OF EXISTING THEORIES

3.1 DEEP PHYSICAL MODEL BASED ON HAMILTONIAN MECHANICS WITH LATENT VARIABLES

In this paper, we consider the combination of deep physical models based on Hamiltonian mechanics and the autoencoder for modeling the equation of motion from images. We call these models deep physical models with latent variables. Note that such models include the deep physical models alone because the encoder and the decoder of the autoencoder can be the identity map.

In deep physical models with latent variables the features extracted by the autoencoder are used as the state variables of the deep physical models. More precisely, we consider typical models that have the architecture shown in Figure 2. To train these models, first, the features of the images are extracted by the encoder. Then the extracted features are sent directly to the decoder to compute the loss function between the output results and the original data. On the other hand, the extracted features are also input into the deep physical model to compute the loss function between the prediction by the deep physical model and the features of the next image computed by the autoencoder.

Obviously, as the deep physical model, arbitrary neural network models for physics can be employed. In this paper, we perform a generalization error analysis and a theoretical analysis on the Hamiltonian interpolation. In the generalization error analysis, we do not place any restriction on the type of the neural network models; for example, the models for energy-dissipative phenomena (e.g., pendulum with friction) can be used. For the latter part, we only consider Hamiltonian systems and assume that deep physical models define a symplectic map.

3.2 SYMPLECTIC INTEGRATORS

As is well known in the field of numerical analysis, symplectic numerical integrators are useful for numerically computing solutions to Hamiltonian equations. Symplectic integrators are designed so that the time evolution map given by the integrators is a symplectic map. For such an integrator, it is known that there exists a so-called shadow Hamiltonian, which defines a Hamiltonian system that approximates the symplectic map corresponding to the integrator. Motivated by this fact, several methods have been proposed to learn the discrete-time Hamilton equation in such a way that the neural network model becomes a symplectic map when learning a time-stepping map.

However, a symplectic map is guaranteed to be related to the Hamilton equation if it is close to the identity map (Benettin & Giorgilli, 1994; Reich, 1999). This is guaranteed for symplectic integrators when the time-step sizes are small enough; however, particularly in the case where the time-step sizes are not so small, it is not necessarily guaranteed that the symplectic map corresponds to the Hamilton equation.

Meanwhile, in the case of learning from video images, the time increments are not always small, and this assumption is generally not satisfied. Therefore, even if the model is constructed to be a symplectic map, it may not be possible to interpolate the images using the Hamilton equation.

3.3 HAMILTONIAN INTERPOLATION OF SYMPLECTIC MAPPINGS

As explained in the above, although any trajectory of a Hamilton equation defines a symplectic map, a symplectic map does not necessarily correspond to a Hamilton equation (Benettin & Giorgilli, 1994). It is known that if a symplectic map is close to the identity map, then there exists a Hamilton equation that approximates this map to a very high degree of accuracy.

Theorem 3.1 (Benettin & Giorgilli (1994)) *For any symplectic mapping ϕ_ε that is analytic and ε -close to the identity, there exists an analytic autonomous Hamiltonian system, H_ε such that the difference between the time-one mapping and ϕ_ε is $O(\varepsilon \exp(-1/\varepsilon))$.*

This theorem has been applied by Reich to evaluate the long-term performance of symplectic numerical integrators (Reich, 1999). In Reich (1999), it is shown that the numerical solutions by the symplectic numerical integrators can be practically regarded as a solution to a Hamilton equation. The Hamiltonian for this Hamilton equation is often called the shadow Hamiltonian. In fact, the symplectic numerical integrators define a symplectic map that is close to the identity map if the time-step size is sufficiently small. Thus, the above theorem is applicable, and the existence of a Hamiltonian system that approximates the symplectic map with an exponentially small error is guaranteed. However, the symplectic map learned by the neural network does not have a parameter corresponding to the time-step size. Therefore, the map may not be sufficiently close to the identity map, and even if the neural network model is symplectic, the existence of the corresponding Hamiltonian system is not obvious.

3.4 SPLITTING METHOD

In the following section, we show that the Hamiltonian interpolation is certainly possible under certain conditions. The proof of this statement relies on the splitting method, which is a technique of numerical analysis. The splitting method is a numerical method for ordinary differential equations, particularly for Hamiltonian systems. Suppose that the Hamiltonian H of a Hamiltonian system is represented by the sum of two Hamiltonians H_1 and H_2 :

$$H(u) = H_1(u) + H_2(u).$$

Then, the time-1 mapping of the Hamiltonian system with respect to H can be approximated by using the time-1 mapping of the Hamiltonian system with respect to H_1 and H_2 . This method is known as the splitting method. As is well known, the difference between these two mappings is shown to be $O(|H_1||H_2|)$ by the Campbell–Baker–Hausdorff lemma (Hairer et al., 2006).

3.5 TOOLS FOR GENERALIZATION ERROR BOUNDS

In statistical learning theory, generalization error bounds are obtained by using the Rademacher complexities or the covering number. See, e.g., Bousquet et al. (2004); Giné & Nickl (2016); Shalev-Shwartz & Ben-David (2014); Steinwart & Christmann (2008) for details.

Definition 3.2 *For a set $V \subset \mathbb{R}^n$,*

$$\mathcal{R}_n(V) := \frac{1}{n} \mathbb{E}_{\sigma \sim \{-1,1\}^n} \sup_{v \in V} \sum_{i=1}^n \sigma_i v_i$$

is called the Rademacher complexity of V .

Lemma 3.3 *Let X, Y be arbitrary spaces, $\mathcal{F} \subset \{f : X \rightarrow Y\}$ be a hypotheses class, and $L : Y \times Y \rightarrow [0, c]$ be a loss function. For a given data set $(x_i, y_i) \in X \times Y$ ($i = 1, \dots, m$), let \mathcal{G} be defined by $\{(x_i, y_i) \in X \times Y \mapsto L(y_i, h(x_i)) \in \mathbb{R} \mid h \in \mathcal{F}, i = 1, \dots, m\}$. 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^m -distributed training data*

$$E[L(Y, h(X))] \leq \frac{1}{m} \sum_{i=1}^m L(y_i, h(x_i)) + 2\mathcal{R}_m(\mathcal{G}) + 3c \sqrt{\frac{2 \ln \frac{4}{\delta}}{m}}$$

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

An estimate of the Rademacher complexity can be obtained by using the covering number.

Definition 3.4 Let V, V' be subsets of \mathbb{R}^n . V is r -covered by V' with respect to the metric function defined by the norm $\|\cdot\|$ if for all $v \in V$ there exists a $v' \in V'$ such that $\|v - v'\| < r$. The covering number $N(r, V, \|\cdot\|)$ of V is the minimum number of the elements of a set that r -covers V .

We also denote $N(r, V, \|\cdot\|)$ by $N(r, V)$ when the norm is clear from the context.

Lemma 3.5 For $V \subset \mathbb{R}^m$ it holds that

$$R(V) \leq \frac{6c}{m} \sum_{k=1}^{\infty} 2^{-k} \sqrt{\log(N(c2^{-k}, V))},$$

where c is the radius of the set V : $c = \min_{v_0 \in V} \max_{v \in V} \|v - v_0\|$. For example, if $\sqrt{\log N(c2^{-k}, V)} \leq \alpha + k\beta$ for some α and β , then $\mathcal{R}_n(V) \leq \frac{6c}{n}(\alpha + 2\beta)$.

Thus, if the covering number is estimated for neural networks models, the bound on the generalization error is obtained. To estimate the covering number, the following lemmas are often used.

Lemma 3.6 Let B_{ρ_B} be a ball with the radius ρ_B in \mathbb{R}^m . Then it holds that $N(r, B_{\rho_B}, \|\cdot\|_2) \leq \left(\frac{2\rho_B}{r} + 1\right)^m$.

Lemma 3.7 For any $V \subset \mathbb{R}^m$, $c > 0$, $v \in \mathbb{R}^m$, $r > 0$, it holds that $N(cr, \{cv + v_0 \mid v \in V\}) \leq N(r, V)$.

Lemma 3.8 Suppose that functions $\phi_i : \mathbb{R} \rightarrow \mathbb{R}$, $i = 1, 2, \dots, m$ are ρ -Lipschitz continuous. For $v \in \mathbb{R}^m$, let $\vec{\phi}(v)$ be defined by

$$\vec{\phi}(v) := (\phi_1(v_1), \dots, \phi_m(v_m)), \quad \vec{\phi} \circ V := \{\vec{\phi}(v) \mid v \in V\}.$$

Then it holds that $N(\rho r, \vec{\phi} \circ V) \leq N(r, V)$, or equivalently, $N(r, \vec{\phi} \circ V) \leq N(\frac{r}{\rho}, V)$.

4 MAIN RESULTS

In the following sections, we perform a generalization error analysis of a model with the structure shown in Figure 2, which combines the autoencoder and a deep physics neural network model, particularly in the case where the model is trained with trajectory data. Note that this model includes the deep physical model alone because the encoder and the decoder in the autoencoder can be the identity map. Hence our generalization error bound can be applied to the deep physical models without the autoencoder.

In addition, if the physics model defines a symplectic map, we also show that this model can be used for interpolating the images while preserving laws of physics even when the time intervals are quite large. By combining the results of the generalization error analysis, Hamiltonian interpolation of near-identity symplectic maps and the error analysis of the splitting method for Hamiltonian systems, we show that the model with this structure can interpolate images while satisfying a Hamiltonian equation even when the time interval is large.

4.1 GENERALIZATION ERROR ANALYSIS OF THE MODELS TRAINED WITH TRAJECTORY DATA

We first show a generalization error bound. We suppose that the model consists of three dense neural networks:

$$f_{\text{NN}} : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_h}, \quad g_{\text{NN}} : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{n_d}, \quad h_{\text{NN}} : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{n_h}.$$

f_{NN} and g_{NN} define an autoencoder: $f_{\text{NN}} \circ g_{\text{NN}} \simeq \text{Id}$. h_{NN} is assumed to be a symplectic map if it is used for Hamiltonian interpolation. As for the neural networks, for simplicity, we consider simple networks with one hidden layer:

$$f_{\text{NN}}(x) = \sigma_{\text{enc}}(A_{\text{enc}}x + b_{\text{enc}}), \quad g_{\text{NN}}(x) = \sigma_{\text{dec}}(A_{\text{dec}}x + b_{\text{dec}}), \quad h_{\text{NN}}(x) = \sigma_{\text{symp}}(A_{\text{symp}}x + b_{\text{symp}}).$$

The extension to deeper networks and/or deep physics neural networks like HNNs is straightforward.

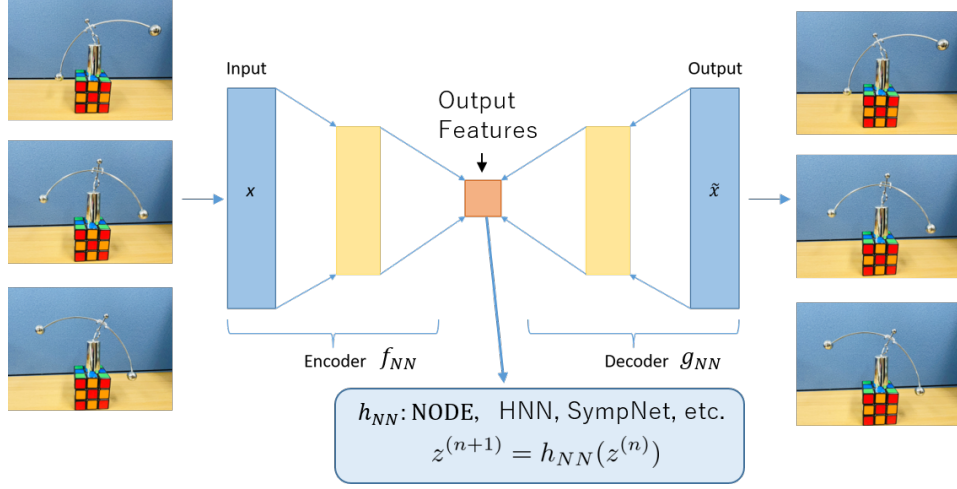


Figure 2: Architecture of the model.

Assumption 4.1 The data set is given as $X = \{x_j^{(1)}, \dots, x_j^{(m_{\text{step}})} \mid j = 1, \dots, m_{\text{orbit}}\} \subset \mathbb{R}^{n_d \times m_{\text{step}} \times m_{\text{orbit}}}$, where $x_j^{(1)}$'s are i.i.d. samples from an unknown distribution D . For each j , $x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(m_{\text{step}})}$ are samples from an orbit starting from $x_j^{(1)}$. m_{orbit} is the number of the orbits. More precisely, we assume that there exists a ρ_ψ -Lipschitz continuous map $\psi : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_d}$ such that for each j and n , $x_j^{(n+1)} = \psi(x_j^{(n)})$. We also assume that $x_j^{(n)}$'s are uniformly bounded so that there exists a compact set $B \subset \mathbb{R}^{n_d}$ such that $x_j^{(n)} \in B$ for any j and n .

We suppose that the model is trained by minimizing the following loss functions:

$$\begin{aligned} & \text{minimize } \frac{1}{m_{\text{orbit}}} \sum_{j=1}^{m_{\text{orbit}}} \mathcal{L}(\{x_j^{(1)}, \dots, x_j^{(m_{\text{step}})}\}), \\ \mathcal{L} &= \mathcal{L}_1 + \mathcal{L}_2, \quad \mathcal{L}_1(\{x_j^{(1)}, \dots, x_j^{(m_{\text{step}})}\}) = \frac{1}{m_{\text{step}}} \sum_n l_1(g_{NN}(f_{NN}(x^{(n)})) - x^{(n+1)}), \\ \mathcal{L}_2(\{x_j^{(1)}, \dots, x_j^{(m_{\text{step}})}\}) &= \frac{1}{m_{\text{step}}} \sum_n l_2(h_{NN}(f_{NN}(x^{(n)})) - f_{NN}(x^{(n+1)})) \end{aligned}$$

where l_1 and l_2 are loss functions.

Assumption 4.2 We assume that there exist a latent variable $z(t) \in \mathbb{R}^{n_h}$, functions $f : \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_h}$ and $g : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{n_d}$ that reconstruct the data $x_j^{(n)}$'s from z . n_h is the dimension of the latent space. We also assume that z satisfies a Hamiltonian equation, and for its existence, we assume that n_h is even. More precisely, we assume that for any $x_j^{(n)} \in X$, it holds that

$$x_j^{(n)} = g(f(x_j^{(n)})), \quad f(x_j^{(n)}) = z((n-1)\Delta t), \quad \frac{dz}{dt} = \begin{pmatrix} O & I \\ -I & O \end{pmatrix} \nabla H(z)$$

with an analytic Hamiltonian H . Δt is the sample time interval.

Assumption 4.3 We assume that the matrices in the neural networks are uniformly bounded and hence the set of the neural networks is given as

$$\begin{aligned} \mathcal{H} &= \{f_{NN}(x) = \sigma_{\text{enc}}(A_{\text{enc}}x + b_{\text{enc}}), \quad g_{NN}(x) = \sigma_{\text{dec}}(A_{\text{dec}}x + b_{\text{dec}}), \\ &h_{NN}(x) = \sigma_{\text{symp}}(A_{\text{symp}}x + b_{\text{symp}}) \mid \|A_{\text{symp}}\| < c_{\text{enc}}, \quad \|A_{\text{symp}}\| < c_{\text{dec}}, \quad \|A_{\text{symp}}\| < c_{\text{symp}}\}. \end{aligned}$$

We also assume that the activation functions σ_{enc} , σ_{dec} and σ_{symp} are analytic and Lipschitz continuous and the Lipschitz constants are bounded by ρ_{enc} , ρ_{dec} , ρ_{symp} , respectively.

Assumption 4.4 For normed spaces X_1, X_2, \dots, X_m , we assume that the norm of the product space $X_1 \times X_2 \times \dots \times X_m$ is defined by the sum of the norms: $\|\cdot\|_{X_1} + \dots + \|\cdot\|_{X_m}$.

Assumption 4.5 The loss function \mathcal{L} can be regarded as a function that maps

$$\left(f_{\text{NN}}(x_j^{(1)}), \dots, f_{\text{NN}}(x_j^{(m_{\text{step}})}), g_{\text{NN}}(x_j^{(1)}), \dots, g_{\text{NN}}(x_j^{(m_{\text{step}})}), h_{\text{NN}}(x_j^{(1)}), \dots, h_{\text{NN}}(x_j^{(m_{\text{step}})}), \right. \\ \left. (x_j^{(1)}, \dots, x_j^{(m_{\text{step}})}) \right)$$

to a real number for each j th orbit. We assume this function is $\rho_{\mathcal{L}}$ -Lipschitz continuous.

The following are the main results of this paper.

Theorem 4.6 Under the above Assumptions, the covering number of the model is bounded by

$$N(r/\rho_{\mathcal{L}}(c_{\text{dec}}\rho_{\text{dec}} + c_{\text{symp}}\rho_{\text{symp}} + 1)(c_{\text{dec}}\rho_{\text{dec}} + c_{\text{symp}}\rho_{\text{symp}} + 1)(m_{\text{step}} \max\{1, \rho_{\psi}^{m_{\text{step}}}\})), \\ B^{m_{\text{orbit}}}).$$

For the proof of this theorem, see Appendix A.1. Since $B^{m_{\text{orbit}}}$ is a compact set in a Euclid space from Assumption 4.1, the estimation of its covering number has been established. For example, its covering number in the 2-norm is estimated by using Lemma 3.6. The covering number in m -norms for m 's other than 2 can be estimated in the same way by using the equivalence of the norms in finite-dimensional spaces. Then, the application of Lemma 3.3 and Lemma 3.5 gives a generalization error bound. If the 2-norm is employed, the Rademacher complexity is shown to be $O(1/\sqrt{m_{\text{orbit}}})$ by using Lemma 3.6. Note that Rademacher complexity mainly depends on the number of trajectories, not on the total number of the data. In fact, in the problem set-up of this study, only the initial values of the trajectories are chosen at random. Therefore, the generalization error depends on the number of the initial values of the trajectories, which means that for the models to be accurate, it is not enough to simply have a large number of data; there should be a large number of randomly sampled trajectory data.

Note also that the estimation of this theorem holds not only for Hamiltonian systems, but also for general physical systems, such as those involving friction. In particular, the physical model does not need to define a symplectic map. In addition, this theorem also holds for data with small time intervals.

Numerical Experiments We performed an experiment to see how the number of trajectories affects the efficiency of learning. The experiments are identical to the 3body problem shown in the original HNNs paper Greydanus et al. (2019). See Greydanus et al. (2019) for the details of the experiments. We used the officially released codes¹ of HNN for this experiment. Regarding the data in this code, the trajectory data are given. More precisely, given the number of time steps m_{step} and the number of trajectories m_{orbit} , numerical solutions of the target differential equation from random initial conditions are computed numerically and m_{step} snapshots are sampled for each orbit. We only changed m_{step} and m_{orbit} and the other settings were unchanged. In the experiments of Greydanus et al. (2019), m_{step} and m_{orbit} are set to $m_{\text{step}} = 20$ and $m_{\text{orbit}} = 5000$. In our experiments, we changed the number of orbits m_{orbit} to 5000, 2500, 1250 and 625. The number of time steps m_{step} is determined so that the number of the total data is unchanged.

The results are shown in Table 2. As expected, the accuracy of the neural network models becomes worse when the number m_{orbit} of the orbits becomes smaller; however, although, theoretically, it was expected that the generalization error would be proportional to $1/\sqrt{m_{\text{orbit}}}$, in practice, when the number of orbits is not large enough, the training error did not decrease sufficiently. This should be because reducing the number of orbits increases the time steps, which makes the time period required to predict each orbit longer and the prediction task more difficult; however, this must be investigated in more detail in future work.

4.2 HAMILTONIAN INTERPOLATION

Second, we investigate whether the interpolated images follow a Hamilton equation or not. An important application of deep physical models is the interpolation of images from videos while

¹<https://github.com/greydanus/hamiltonian-nn>

Table 2: Training results of Hamiltonian neural networks for the 3-body problem. Although the total number of data is constant, the accuracy improves as the number of trajectories increases.

| m_{orbit} | m_{step} | TRAINING ERROR | TEST ERROR | DIFFERENCE |
|--------------------|-------------------|---|---|---------------------|
| 625 | 160 | $8.3945\text{E}+01 \pm 1.4453\text{E}+02$ | $2.3640\text{E}+01 \pm 1.5563\text{E}+01$ | $6.0305\text{E}+01$ |
| 1250 | 80 | $3.9873\text{E}+00 \pm 2.8671\text{E}+00$ | $9.9912\text{E}+00 \pm 1.5279\text{E}+00$ | $6.0039\text{E}+00$ |
| 2500 | 40 | $1.6950\text{E}+00 \pm 1.4612\text{E}+00$ | $2.2142\text{E}+00 \pm 1.5869\text{E}+00$ | $0.5192\text{E}+00$ |
| 5000 | 20 | $8.5064\text{E}-02 \pm 3.0044\text{E}-02$ | $4.6424\text{E}-01 \pm 4.2729\text{E}-01$ | $3.7918\text{E}-01$ |

preserving the laws of physics. As an application of the error analysis, we show that it is certainly possible to interpolate the images from videos so that the laws of physics are preserved, provided that the generalization error is sufficiently small.

Assumption 4.7 We assume that h_{NN} is symplectic.

Assumption 4.8 For the loss function l_2 , we assume that the following holds: if the expectation value of l_2 is sufficiently small, then the map determined by h_{NN} is close enough to the time Δt map $\psi_{\Delta t}$ of the Hamilton equation satisfied by the latent variable z . More precisely, we assume that $\psi_{\Delta t}^{-1}$ exists, and that there exists a constant c_{l_2} such that the operator norm of the difference between the two operators is bounded by

$$\|h_{\text{NN}} \circ \psi_{\Delta t}^{-1} - \text{Id}\| \leq c_{l_2} E[\mathcal{L}_2].$$

For Assumption 4.8, for example, a sufficiently high order Sobolev norm can be used as the loss function. In fact, if the order is sufficiently large than n_h , the Sobolev embedding theorem gives a bound of the error in L^∞ norm. Thus, since the error is uniformly bounded in the domain of h_{NN} , the map defined by h_{NN} is close enough to the time Δt map of the Hamilton equation satisfied by the latent variable z . Besides, the existence of $\psi_{\Delta t}^{-1}$ is not a strong assumption because Hamiltonian systems are typically invertible. On the other hand, note that we do not place the assumption that the time interval of the images to be interpolated is sufficiently small. Rather, we suppose that the time intervals may be quite large.

Theorem 4.9 Suppose that the loss function \mathcal{L} for the training data satisfies $\mathcal{L} \leq \varepsilon_1$. Then, under the above Assumptions, for arbitrary $\varepsilon_2 > 0$, there exists $0 < \delta < 1$ such that with probability $1 - \delta$ there is an Hamiltonian $\hat{H} : \mathbb{R}^{n_h} \rightarrow \mathbb{R}$ that defines Hamiltonian system of which the time-1 map is an $O(c_{l_2}(\varepsilon_1 + 2\mathcal{R}_{m_{\text{orbit}}}) + \varepsilon_2)$ approximation to the symplectic map h_{NN} .

For the proof, see Appendix A.2. Roughly, as seen in Figure 1, first, the map h_{NN} is decomposed into a composition of two maps that are related to Hamiltonian systems, and then the splitting method is applied. Theorem 4.9 means that with a certain probability the trained model admits a Hamiltonian interpolation which is close to the true Hamiltonian system.

5 CONCLUSION

Recently, neural network methods for learning the equations of motion from observational data have attracted much attention. Although the theoretical analysis of such models has been gradually developed, a generalization error for such models trained with trajectory data has not yet been investigated. To the best of our knowledge, this paper is the first work on the error analysis for such a case. Our analysis shows an error bound that is dependent on the number of trajectories, not on the number of the whole data.

A possible application of these models is to interpolate images while preserving the laws of physics. Such interpolation is called Hamiltonian interpolation. In this paper, we also consider this application and show that under certain assumptions Hamiltonian interpolation is certainly possible by combining the autoencoder and a deep physical model that defines a symplectic map.

REFERENCES

- Giancarlo Benettin and Antonio Giorgilli. On the hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms. *J. Stat. Phys.*, 74(5): 1117–1143, March 1994.
- Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch (eds.), *Advanced Lectures on Machine Learning*, pp. 169–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018. URL <http://arxiv.org/abs/1806.07366>.
- Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Kam theory meets statistical learning theory: Hamiltonian neural networks with non-zero training loss, 2021a. URL <https://arxiv.org/abs/2102.11923>.
- Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Neural symplectic form: Learning hamiltonian equations on general coordinate systems. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021b. URL <https://openreview.net/forum?id=4h4oqp-ATxb>.
- Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks, 2020. URL <https://arxiv.org/abs/2003.04630>.
- Rembert Daems, Jeroen Taets, Francis Wyffels, and Guillaume Crevecoeur. Keyld: Learning constrained lagrangian dynamics in keypoint coordinates from images. *ArXiv*, abs/2206.11030, 2022.
- Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge University Press, 2016.
- Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *CoRR*, abs/1906.01563, 2019. URL <http://arxiv.org/abs/1906.01563>.
- Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. 2006.
- Florian Hofherr, Lukas Koestler, Florian Bernard, and Daniel Cremers. Neural implicit representations for physical parameter inference from a single video. *ArXiv*, abs/2204.14030, 2022.
- Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram S. Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jérôme Parent-Lévesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. *ArXiv*, abs/2104.02646, 2021.
- Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, and George Em Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems, 2020. URL <https://arxiv.org/abs/2001.03750>.
- Pengzhan Jin, Zhen Zhang, Ioannis G Kevrekidis, and George Em Karniadakis. Learning poisson systems and trajectories of autonomous systems via poisson neural networks. *IEEE Trans Neural Netw Learn Syst*, PP, February 2022.
- Mohammad Asif Khan and Amos J. Storkey. Hamiltonian operator disentanglement of content and motion in image sequences. *ArXiv*, abs/2112.01641, 2021.
- Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Aizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.*, 24(89):1–97, 2023.
- P. Langley. Crafting papers on machine learning. In Pat Langley (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

- Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *ArXiv: 2010.08895*, 2020.
- Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021.
- J. Mason, Christine Allen-Blanchette, Nick Zolman, Elizabeth Davison, and Naomi Ehrich Leonard. Learning interpretable dynamics from images of a freely rotating 3d rigid body. *ArXiv*, abs/2209.11355, 2022.
- Takashi Matsubara, Ai Ishikawa, and Takaharu Yaguchi. Deep energy-based modeling of discrete-time physics, 2019. URL <https://arxiv.org/abs/1905.08604>.
- M Raissi, P Perdikaris, and G E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, February 2019.
- Sebastian Reich. Backward error analysis for numerical integrators. *SIAM J. Numer. Anal.*, 36(5):1549–1570, January 1999.
- Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Peter Deisenroth. Variational integrator networks for physically structured embeddings. 2019. doi: 10.48550/ARXIV.1910.09349. URL <https://arxiv.org/abs/1910.09349>.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, May 2014.
- Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, September 2008.
- Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. September 2019.
- Zhen Zhang, Yeonjong Shin, and George Em Karniadakis. GFINNs: GENERIC formalism informed neural networks for deterministic and stochastic dynamical systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229), jun 2022. doi: 10.1098/rsta.2021.0207. URL <https://doi.org/10.1098/rsta.2021.0207>.
- Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: Learning hamiltonian dynamics with control. September 2019.