

Data-Driven Identification of Quadratic Symplectic Representations of Nonlinear Hamiltonian Systems

Süleyman Yıldız* Pawan Goyal** Thomas Bendokat† Peter Benner†‡

*Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.
Email: yildiz@mpi-magdeburg.mpg.de, ORCID: 0000-0001-7904-605X

**Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.
Email: goyal@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3072-7780

†Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.
Email: bendokat@mpi-magdeburg.mpg.de, ORCID: 0000-0002-0671-6291

†‡Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.
Email: benner@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3362-4103

††Otto von Guericke University, Universitätsplatz 2, 39106 Magdeburg, Germany
Email: peter.benner@ovgu.de

Abstract: We present a framework for learning Hamiltonian systems using data. This work is based on the lifting hypothesis, which posits that nonlinear Hamiltonian systems can be written as nonlinear systems with cubic Hamiltonians. By leveraging this, we obtain quadratic dynamics that are Hamiltonian in a transformed coordinate system. To that end, for given generalized position and momentum data, we propose a methodology to learn quadratic dynamical systems, enforcing the Hamiltonian structure in combination with a symplectic auto-encoder. The enforced Hamiltonian structure exhibits long-term stability of the system, while the cubic Hamiltonian function provides relatively low model complexity. For low-dimensional data, we determine a higher-order transformed coordinate system, whereas for high-dimensional data, we find a lower-order coordinate system with the desired properties. We demonstrate the proposed methodology by means of both low-dimensional and high-dimensional nonlinear Hamiltonian systems.

Keywords: Cubic Hamiltonian, quadratic Hamiltonian systems, lifting principle for dynamical systems, structure-preserving model order reduction, symplectic auto-encoder

Novelty statement:

- Inspired by quadratic lifting, allowing to rewrite nonlinear systems as quadratic systems in a lifted coordinate system, we discuss a lifting principle for nonlinear Hamiltonian systems.
- We propose a data-driven approach to learning a quadratic Hamiltonian coordinate system by means of symplectic auto-encoders so that
 - the dynamics in the learned coordinate system can be given by a quadratic system, and
 - the underlying Hamiltonian function is cubic.
- For high-dimensional data, we discuss learning a reduced coordinate system so that the above goals are achieved. This then aligns with non-intrusive model-order reduction by nonlinear projection.
- By means of several examples, including high-dimensional ones, we demonstrate the proposed methodology.

1. Introduction

Hamiltonian dynamics are ubiquitous as a powerful mathematical tool in modeling complex physical dynamical systems [1]. Classically, they are used in topics ranging from celestial mechanics [2] over fluid mechanics [3] to Schrödinger equations in quantum mechanics [4]. The coordinates in which Hamiltonian systems operate are split into generalized positions and momenta, which need to be identified from given data in order to fit a physical model to observations.

The construction of models that can accurately capture and predict the dynamics of highly complex systems has been of interest for several decades if not centuries; see, e.g., [5] and references therein. Recently, the powerful approximation capabilities of neural networks have brought researchers in many fields closer to understanding complicated systems. Neural networks have been successfully studied for predicting complex dynamical systems [6, 7], improving turbulence models [8, 9], classifying time series [10], and studying differential equations (DEs) [11, 12]. To exploit the long-term stability properties of Hamiltonian systems, neural networks are used to learn the energy functions [13–16] rather than dynamical systems.

In this work, we are interested in learning quadratic Hamiltonian systems explaining given trajectory data from two different perspectives: lifting transformations [17], and nonlinear symplectic model order reduction [18] with weak enforcement of the transformation to be symplectic. In fact, we learn the quadratic Hamiltonian systems directly from data, without needing to resort to the original dynamical equations.

In short, given data from a Hamiltonian system, we want to learn the dynamics in a structure-preserving way, while achieving low model complexity and having the option to reduce the dimension for high-dimensional data. This is achieved via structure-preserving auto-encoders and modeling of the dynamics with a quadratic Hamiltonian system.

In [17], a unified approach, namely lifting transformations, is used to approximate general nonlinear systems. In the case where the dynamical system is known, one can manually design lifted variables. However, lifting the dynamical system does not necessarily lead to a Hamiltonian system. Therefore, we weakly force the lifting transformations to be symplectic by exploiting symplectic embeddings and strictly enforce the Hamiltonian structure of the dynamics equations.

The second application of our approach lies in dimensionality reduction of Hamiltonian systems. Learning reduced-order models for Hamiltonian systems comes with some practical challenges. Without enforcing preservation of the Hamiltonian structure in the reduced-order model, it can quickly lose accuracy [19]. One established approach to preserve the symplectic structure is by using linear symplectic projections, i.e., proper symplectic decomposition [19, 20], but for Hamiltonian systems with slow decay of the Kolmogorov- n -width, this approach might not be feasible. Furthermore, for non-linear Hamiltonian functions, hyperreduction methods like the Symplectic Discrete Empirical Interpolation Method (SDEIM) [19] are needed for efficient computability of the reduced-order model.

In [18], a reduction by a non-linear structure-preserving auto-encoder is studied, addressing this problem. We take a similar approach and use a structure-preserving auto-encoder to map data from a Hamiltonian system to a learned *quadratic* Hamiltonian system, thereby reducing model complexity considerably. The simple quadratic structure allows for the direct learning of the Hamiltonian system, without having to learn the Hamiltonian function and without the need of calculating its gradient. For the purpose of learning reduced dynamics, similar as in MOR, we show that this quadratic Hamiltonian system can be of much lower dimension than the original full order model. Learning the quadratic Hamiltonian system from data has the further advantage that no hyper-reduction methods are needed for nonlinear systems, and the reduced-order model can thus be efficiently computed. Furthermore, as our approach learns the reduced dynamics directly, we do not need to take the gradient through the auto-encoder to simulate the learned models.

The recent preprint [21] can be seen as a complementary approach to our method for reducing the order of Hamiltonian systems. While we learn a quadratic Hamiltonian system with a general non-linear symplectic auto-encoder, in [21] two different versions of quadratic symplectic auto-encoders are studied, which are then used for model order reduction leading to a general non-linear Hamiltonian system. Moreover, we learn the reduced dynamics directly from data, while [21] studies model order reduction, i.e., resorting to the Hamiltonian of the full-order model. In future work, a combination of both approaches seems worthwhile.

The paper is structured as follows. In Section 2, we introduce the necessary mathematical background to embed Hamiltonian systems in a structure-preserving way into a higher-dimensional space and define quadratic Hamiltonian systems. In Section 3 we describe the auto-encoder structure to lift Hamiltonian

systems. In Section 4 we adapt the theory to learn low-dimensional quadratic representations of high-dimensional data in a structure-preserving way. In Section 5 we show the applicability of the approach, for low-dimensional systems in Subsection 5.1 and for structure preserving reduction of high-dimensional systems in Subsection 5.2. Section 6 concludes the paper. Implementation details can be found in Appendix A.

2. Background

In this section, we provide the necessary theoretical background needed for the derivation of our learning approach for Hamiltonian systems.

2.1. Hamiltonian Systems and Symplectic Embedding

The governing equations of Hamiltonian systems are Hamilton's equations, namely

$$\dot{x}(t) = J_{2n} \nabla_x \mathcal{H}(x(t)) \in \mathbb{R}^{2n}, \quad (2.1)$$

where $x = (q, p) \in \mathbb{R}^{2n}$ with q and p being generalized positions and momenta, respectively,

$$J_{2n} := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n},$$

and ∇_x denotes the gradient with respect to x . Moreover, we consider an initial condition $x(0) = x_0 = (q_0, p_0) \in \mathbb{R}^{2n}$. The Hamiltonian function $\mathcal{H}: \mathbb{R}^{2n} \rightarrow \mathbb{R}$ describes the energy of the system and is preserved along the solution trajectories. Next, we discuss the definition of a symplectic embedding, which plays an important role in our later discussions.

Definition 1 (Symplectic Embedding for Vector Spaces). *A symplectic embedding of \mathbb{R}^{2n} into \mathbb{R}^{2N} is a homeomorphism $\psi: \mathbb{R}^{2n} \rightarrow \psi(\mathbb{R}^{2n}) \subset \mathbb{R}^{2N}$ for which the Jacobian $d\psi_x \in \mathbb{R}^{2N \times 2n}$ fulfills*

$$(d\psi_x)^T J_{2N} d\psi_x = J_{2n} \quad (2.2)$$

at every $x \in \mathbb{R}^{2n}$.

It is immediate to see that a symplectic embedding is a smooth embedding in the sense of differential geometry [22, Section 22, p. 568], as the Jacobian has full rank at every point. The Jacobian is therefore injective and ψ is an immersion. This furthermore implies $N \geq n$. Therefore, a symplectic embedding is also called a *symplectic lifting*.

Proposition 1 (Equivalent Embedded System). *Let $\psi: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2N}$ be a symplectic embedding and define $z_0 := \psi(x_0) \in \mathbb{R}^{2N}$. Then, the system (2.1) is equivalent to the embedded system*

$$\dot{z}(t) = J_{2N} \nabla_z \mathcal{H}(\psi^{-1}(z(t))), \quad (2.3)$$

i.e., the solution of the differential equation fulfills $z(t) = \psi(x(t))$ for all $t \in [0, \infty)$.

Proof. For any $z \in \psi(\mathbb{R}^{2n})$, it holds with the chain rule that

$$\begin{aligned} J_{2N} \nabla_z \mathcal{H}(\psi^{-1}(z)) &= J_{2N} (d(\mathcal{H} \circ \psi^{-1})_z(z))^T = J_{2N} (d\psi^{-1}_z)^T (d\mathcal{H}_x(\psi^{-1}(z)))^T \\ &= J_{2N} (d\psi^{-1}_z)^T \nabla_x \mathcal{H}(\psi^{-1}(z)) = d\psi_x J_{2n} \nabla_x \mathcal{H}(\psi^{-1}(z)). \end{aligned}$$

As $z(t) := \psi(x(t))$ implies $\dot{z}(t) = d\psi_{x(t)} \dot{x}(t) = d\psi_{x(t)} J_{2n} \nabla_x \mathcal{H}(x(t))$, the claim follows. \square

As ψ is a symplectic lifting, the system for z is called a *symplectic lifting of the system for x* .

2.2. Quadratic Symplectic Representations

There are many possibilities to construct a symplectic embedding of a nonlinear Hamiltonian system. However, in this work, we are seeking to identify a particular higher dimensional or lifted space so that a quadratic system can describe the dynamics in the lifted space. Moreover, the Hamiltonian in the lifted

space is a cubic polynomial function. To briefly describe the lifting procedure, we consider the system of ODEs

$$\dot{x} = f(x(t)), \quad x \in \mathbb{R}^n. \quad (2.4)$$

The quadratic lifting transformation [17, 23–25] can be obtained by defining a transformation $z(t) = \psi(x(t)) \in \mathbb{R}^N$ for $N \geq n$ such that the transformed system (2.4) satisfies

$$\dot{z} = \mathcal{A} + \mathcal{B}z + \mathcal{C}z \otimes z \in \mathbb{R}^N. \quad (2.5)$$

We illustrate the quadratic lifting for nonlinear systems by means of a nonlinear oscillator example.

Example 1 (Nonlinear Oscillator). *Consider the nonlinear (an-harmonic) oscillator [26] with the Hamiltonian $\mathcal{H}(q, p) = \frac{p^2}{2} + \frac{q^2}{2} + \frac{q^4}{4}$. The associated governing equations for this oscillator are given by*

$$\begin{aligned} \dot{q} &= p, \\ \dot{p} &= -(q + q^3). \end{aligned} \quad (2.6)$$

We demonstrate the lifting transformation by introducing the variables $w_1 = q$, $w_2 = p$, and $w_3 = q^2$. With the new variable w_3 , the equations of motion for the oscillator (2.6) can be written as

$$\begin{aligned} \dot{w}_1 &= w_2, \\ \dot{w}_2 &= -(w_1 + w_1 w_3), \\ \dot{w}_3 &= 2w_1 w_2, \end{aligned} \quad (2.7)$$

which is a quadratic system. Moreover, one can also define an inverse mapping from (w_1, w_2, w_3) to (q, p) . However, it is easy to note that the system in (2.7) is not a Hamiltonian system in canonical coordinates since it is odd-dimensional. We further note that even introducing new variables to make the lifted system (2.7) even dimensional does not necessarily result in a Hamiltonian system.

Notably, the theory of generating functions can be used to construct quadratic Hamiltonian systems. For a detailed overview of generating functions, we refer to the book [27]. To illustrate this for the nonlinear oscillator, we suppose $\hat{p} = q^2$. Then, using a generating function of type 1, one can find $F_1 = -\hat{q}q^2$, so that $p = -2\hat{q}q$, implying $p = -2\hat{q}\hat{p}^{1/2}$ and $q = \hat{p}^{1/2}$. The new Hamiltonian with new variables becomes $\hat{\mathcal{H}} = 2\hat{q}^2\hat{p} + \frac{\hat{p}}{2} + \frac{\hat{p}^2}{4}$, which is cubic; hence, the underlying dynamics are given by a quadratic system.

Inspired by the above example, in this work, we seek to identify a symplectic space to lift to. The desired properties can be achieved when the lifted system (2.5) satisfies (2.3) with a symplectic lifting ψ fulfilling (2.2). For this, we first define quadratic Hamiltonian systems for our reference.

Definition 2 (Quadratic Hamiltonian System). *A quadratic Hamiltonian system is a Hamiltonian system (2.1) for which the Hamiltonian function is cubic, i.e.,*

$$\mathcal{H}(x) = A^T x + B^T(x \otimes x) + C^T(x \otimes x \otimes x),$$

where $A \in \mathbb{R}^{2n}$, $B \in \mathbb{R}^{(2n)^2}$, $C \in \mathbb{R}^{(2n)^3}$, and \otimes denotes the Kronecker product.

The simple structure of quadratic Hamiltonian systems allows enforcing the Hamiltonian condition directly onto the system, without having to compute the gradient of the Hamiltonian function.

Proposition 2. *A quadratic system of ODEs*

$$\dot{x} = \mathcal{A} + \mathcal{B}x + \mathcal{C}(x \otimes x) \in \mathbb{R}^{2n}, \quad (2.8)$$

where $\mathcal{A} \in \mathbb{R}^{2n}$, $\mathcal{B} \in \mathbb{R}^{2n \times 2n}$ and $\mathcal{C} \in \mathbb{R}^{2n \times (2n)^2}$, is a quadratic Hamiltonian system if and only if $J_{2n}^T \mathcal{B}$ is a symmetric matrix and there is a symmetric tensor $\mathcal{T} \in \mathbb{R}^{2n \times 2n \times 2n}$ for which

$$\mathcal{T}_u(x \otimes x) = J_{2n}^T \mathcal{C}(x \otimes x)$$

holds for all $x \in \mathbb{R}^{2n}$, where $\mathcal{T}_u \in \mathbb{R}^{2n \times (2n)^2}$ is the unfolding of \mathcal{T} by frontal slices.

Proof. By definition, (2.8) is a quadratic Hamiltonian system if and only if there is a cubic function $\mathcal{H}(x) = A^T x + B^T x \otimes x + C^T x \otimes x \otimes x$ such that $\dot{x} = J_{2n} \nabla_x \mathcal{H}(x)$, which is equivalent to

$$\nabla_x \mathcal{H}(x) = \nabla_x (A^T x + B^T(x \otimes x) + C^T(x \otimes x \otimes x)) = J_{2n}^T (\mathcal{A} + \mathcal{B}x + \mathcal{C}x \otimes x).$$

Since there is a bijection between homogenous polynomials and symmetric tensors [28, p. 6], the claim follows. \square

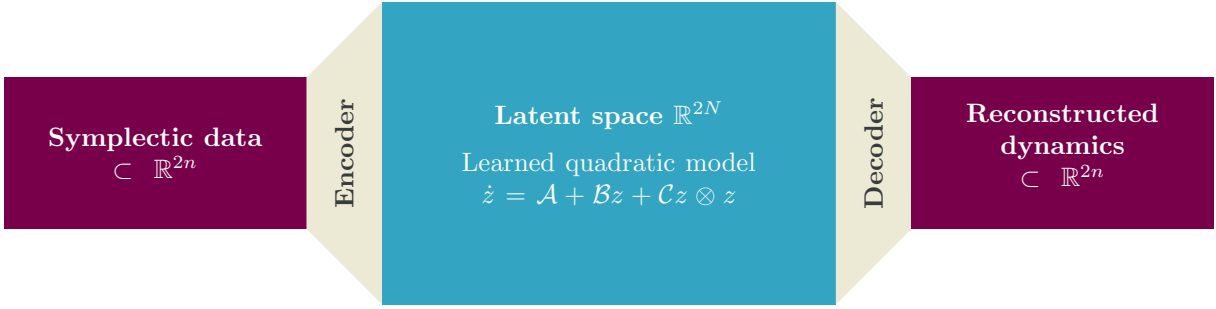


Figure 3.1: The auto-encoder structure of the symplectic lifting method. Here, the encoder $\psi : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2N}$ is weakly enforced to be a symplectic mapping and the quadratic system is enforced to be Hamiltonian.

For many smooth nonlinear systems, there exist guaranteed liftings which allow us to rewrite nonlinear systems as quadratic systems, see, e.g., [23, 24]. However, there is currently no established result ensuring the existence of a symplectic lifting for nonlinear Hamiltonian systems to higher dimensions where the dynamics can be represented by quadratic Hamiltonian systems. Exploring this aspect remains an intriguing theoretical endeavor for future research. In this work, however, we hypothesize the existence of such a system and focus on learning such a symplectic lifting/embedding by means of suitable optimization problems, which we discuss next.

3. Learning the Lifted Quadratic Symplectic Representation

Here, we describe our methodology to learn a symplectic lifting to map from a given canonical Hamiltonian system to a quadratic Hamiltonian system, which we visualize in Figure 3.1. The first ingredient to it is to define lifted coordinates $z(t)$ using a classical auto-encoder loss as follows:

$$\mathcal{L}_{\text{encdec}} = \|x(t) - \phi(\psi(x(t)))\|, \quad (3.1)$$

where $\psi(x(t)) = z(t) = (\hat{q}(t), \hat{p}(t))$ and $\phi(z(t)) = \tilde{x}(t) = (q(\hat{q}(t)), p(\hat{p}(t))) \approx x(t)$. However, the mapping obtained through (3.1) does not necessarily yield a symplectic mapping. To get a symplectic map, we use (2.2) and define a symplectic loss as follows:

$$\mathcal{L}_{\text{symp}} = \|(\text{d}\psi_x)^T J_{2N} \text{d}\psi_x - J_{2n}\|. \quad (3.2)$$

Furthermore, we assume that time derivatives of states are accessible. Thus, we compute the time derivatives of the lifted space z using the chain-rule. Hence, we add the following term in the loss function:

$$\begin{aligned} \mathcal{L}_{z\dot{x}} &= \|\text{d}\psi_{x(t)}\dot{x}(t) - J_{2N}\nabla_z\mathcal{H}(\psi^{-1}(z(t)))\| \\ &= \|\text{d}\psi_{x(t)}\dot{x}(t) - (\mathcal{A} + \mathcal{B}z(t) + \mathcal{C}z(t) \otimes z(t))\| \end{aligned} \quad (3.3)$$

with $z = \psi(x)$. Finally, to obtain a quadratic Hamiltonian system, we combine all these losses defined in (3.1)–(3.3). Hence we have the total loss as a weighted sum of these loss functions, given by

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{encdec}} + \lambda_2\mathcal{L}_{\text{symp}} + \lambda_3\mathcal{L}_{z\dot{x}}, \quad (3.4)$$

where $\lambda_{\{1,2,3\}}$ are hyper-parameters. The details of the implementation and auto-encoders are given in Appendix A. Finally, we optimise all parameters in (3.4) at the same time.

We remark that we do not enforce the homeomorphism property, but only enforce (2.2), i.e., the condition that the encoder is a symplectic immersion and therefore locally invertible, and that the encoder is invertible on the training data.

4. Low-dimensional Quadratic Symplectic Representation of High-dimensional Data

Thus far, we have discussed how nonlinear Hamiltonian systems can be lifted to higher dimensional quadratic symplectic systems and how they can be learned by means of data. However, there are many

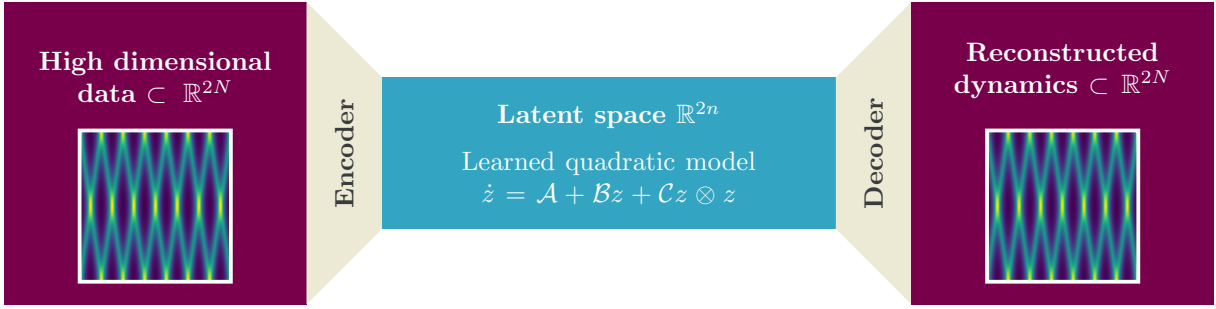


Figure 4.1: The auto-encoder structure of the symplectic reduction method. Here, the decoder $\phi : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2N}$ is weakly enforced to be a symplectic mapping and the quadratic system is enforced to be Hamiltonian.

Hamiltonian systems which are high-dimensional, specially coming from partial differential equations. Furthermore, it is known that high dimensional dynamic data often evolve in a lower-dimensional subspace. Therefore, in these cases, we aim to learn lower-dimensional coordinates for high-dimensional data so that they are not only symplectic but can also be used to describe the dynamics of the high-dimensional system, as depicted in Figure 4.1.

However, there is a subtlety compared to the method discussed in the previous section. It is worth noting that (3.2) will weakly enforce symplecticity in the case of symplectic lifting. On the other hand, for high-dimensional data, the quadratic system is of lower dimension than the original data; thus, we actually need to enforce that the *decoder* ϕ , and not the encoder ψ , of the auto-encoder is a symplectic embedding from the quadratic model to the original high-dimensional system.

Since the high-dimensional data, particularly coming from partial differential equations, have a lot of spatial coherency, we make use of a deep convolutional auto-encoder (DCA), which is computationally efficient. Moreover, to enforce the symplecticity condition in the loss function, we use a weakly symplectic deep convolutional auto-encoder with essentially the same conditions as in [18, Section 3.3]. This means that in the symplectic reduction, instead of (3.2), we use the following loss terms for symplectic loss:

$$\tilde{\mathcal{L}}_{\text{symp}} = \|(\text{d}\phi_z)^T J_{2N} \text{d}\phi_z - J_{2n}\|. \quad (4.1)$$

As we have switched the role of n and N in the reduction case, (3.3) is replaced by

$$\begin{aligned} \tilde{\mathcal{L}}_{\dot{z}\dot{x}} &= \|\text{d}\psi_{x(t)}\dot{x}(t) - J_{2n}\nabla_z\mathcal{H}(\psi^{-1}(z(t)))\| \\ &= \|\text{d}\psi_{x(t)}\dot{x}(t) - (\mathcal{A} + \mathcal{B}z(t) + \mathcal{C}z(t) \otimes z(t))\|. \end{aligned} \quad (4.2)$$

The loss of the auto-encoder $\mathcal{L}_{\text{encdec}}$ is given by (3.1) as in the lifting case. Hence, the total loss is calculated via

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{encdec}} + \lambda_2 \tilde{\mathcal{L}} + \lambda_3 \tilde{\mathcal{L}}_{\dot{z}\dot{x}}, \quad (4.3)$$

which is then used to learn a suitable embedding.

5. Numerical Experiments

In this section, we examine the performance of the proposed methodology in two scenarios: low-dimensional dynamical systems and high-dimensional dynamical systems. For the low-dimensional case, we investigate three different examples: the simple pendulum, an an-harmonic oscillator, and the Lotka-Volterra equations. For the high-dimensional case, we study the linear wave and nonlinear Schrödinger equations. All the experiments are done using PyTorch on a machine with an Intel® Core™ i5-12600K CPU and NVIDIA RTX™ A4000(16GB) GPU. To preserve the symplectic structure after time discretization, we have used the implicit midpoint rule as time integrator. In the case of symplectic lifting for all low-dimensional examples, we set the dimension of the latent space—for which the dynamics are quadratic and have a constant cubic Hamiltonian—to four. In the case of symplectic reduction we set the dimension of the latent space of the linear wave equation to four and of the nonlinear Schrödinger equation to two. All other hyperparameter settings and neural network architectures are listed in detail in Appendix A for each example.

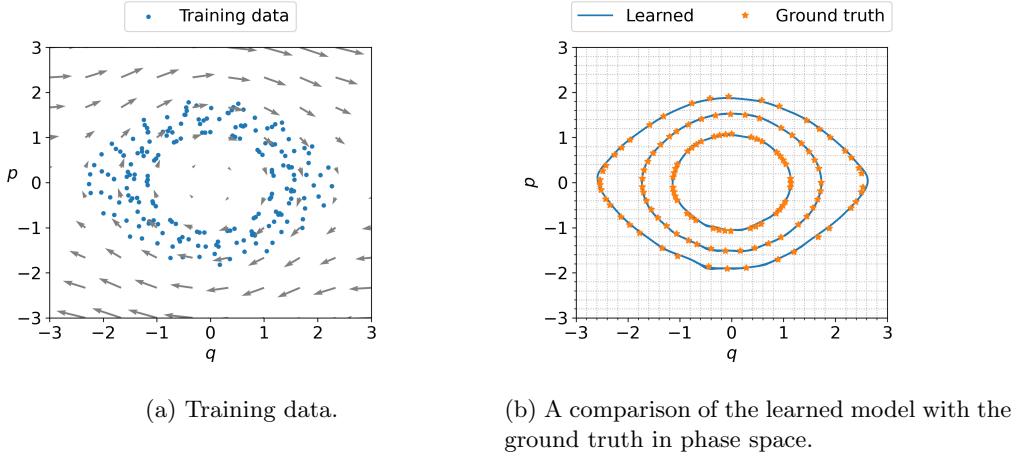


Figure 5.1: Nonlinear pendulum: the plot (a) shows training data of the pendulum example in phase space, and the plot (b) shows a comparison of the learned model with the ground truth in phase space with three random initial conditions.

5.1. Low-dimensional Systems

Here, we discuss learning dynamical systems using low-dimensional data by means of three examples.

5.1.1. Nonlinear Pendulum

Our first example of low-dimensional dynamics is a frictionless pendulum. Pendulums are non-linear oscillators and Hamiltonian systems, making them challenging to learn solely from data. The Hamiltonian for the system can be given by

$$\mathcal{H}(q, p) = 2mgl(1 - \cos(q)) + \frac{l^2 p^2}{2m}, \quad (5.1)$$

where g represents the gravitational constant, l denotes the length of the pendulum, and m denotes the mass. For simplicity, we set the mass of the pendulum to $m = 1.0$, its length to $l = 1.0$, and the gravitational constant to $g = 0.5$. Consequently, we can express the governing equations that define the evolutions of p and q as follows:

$$\begin{bmatrix} \dot{p}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} -\sin(q(t)) \\ p(t) \end{bmatrix}. \quad (5.2)$$

To generate the training dataset, we consider initial conditions for variables p and q within the range of $[-2, 2] \times [-2, 2]$, encompassing the transition from linear to nonlinear dynamics in the system. However, to avoid a complete circle described by the pendulum, we select initial conditions from the range with energy $\mathcal{H}(q, p) < 2$. We consider 10 random initial conditions and take 50 equidistant data points in the time interval $[0, 10]$. We have pictorially shown the training data in [Figure 5.1a](#).

Next, we learn latent variables \hat{p} and \hat{q} with our desired objective, which is that the dynamics of the latent variables can be given by a quadratic system with a cubic Hamiltonian. Moreover, the latent variables are learned by means of an encoder, and the quantities-of-interest, namely p and q , are identified using a decoder, which maps the latent variables to $p(\hat{q}, \hat{p})$ and $q(\hat{q}, \hat{p})$. With the training configuration given in [Appendix A](#), we first demonstrate the learned dynamics in the phase space for three random test initial conditions in [Figure 5.1b](#) for the pendulum example, where the figure shows that the learned system is stable and orbiting at the same energy level as the ground truth model. Furthermore, in [Figure 5.2](#), we compare time-domain simulations of the identifying model with the ground truth model for a random initial condition that is different from the training set. [Figure 5.2](#) shows that the learned model is not only good at capturing the dynamics in a test case but also stable and accurate for long-time integration, on a time interval larger than the training interval $[0, 10]$.

In [Figure 5.3](#), we plot the learned and canonical Hamiltonians, demonstrating that all Hamiltonians remain constant over time with minor oscillations. Notably, the learned Hamiltonian closely aligns with the canonical coordinates, even without the need for additional constraints in the optimization process, such as weakly enforcing the initial Hamiltonian value.

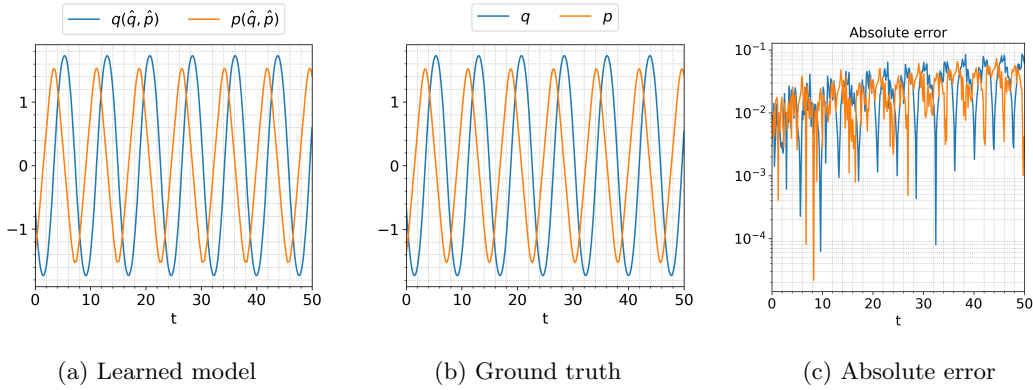


Figure 5.2: Nonlinear pendulum: A comparison of the learned model with the ground truth model for a random test condition.

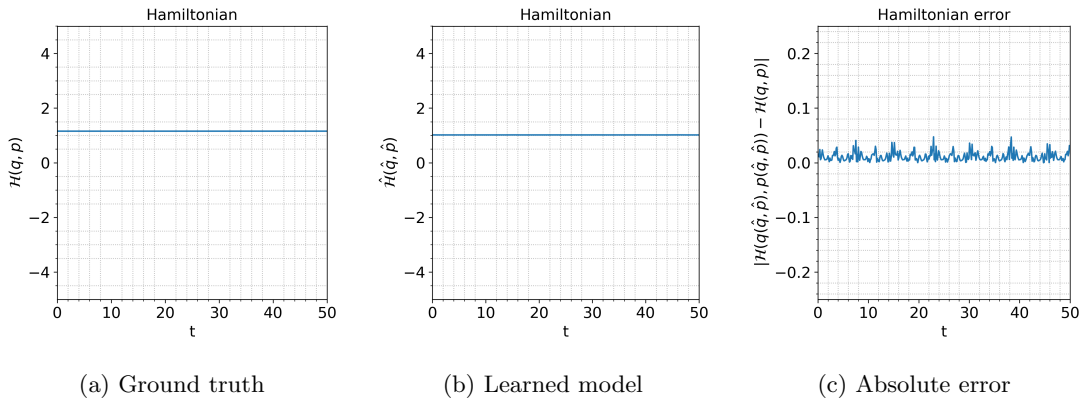


Figure 5.3: Nonlinear pendulum: A comparison of the Hamiltonian in canonical coordinates for the ground truth model $\mathcal{H}(q, p)$, the learned Hamiltonian $\hat{\mathcal{H}}(\hat{q}, \hat{p})$ in the latent space, and the difference between the ground truth model and the learned model in the original space $\mathcal{H}(q(\hat{q}, \hat{p}), p(\hat{q}, \hat{p}))$ along time using a random test initial condition.

5.1.2. Lotka–Volterra Equations

Our second example of a low-dimensional system is the Lotka-Volterra system [29]. The Lotka-Volterra model is a well-known mathematical model used to describe predator-prey populations' dynamics. This model has been extensively applied to study the population dynamics of diverse species across various ecosystems. Moreover, it is an example of a system with an underlying Hamiltonian structure with Hamiltonian

$$\mathcal{H}(q, p) = p - e^p + 2q - e^q.$$

To learn the dynamics of the Lotka-Volterra equations, we constructed a training set with 10 trajectories. These trajectories were simulated up to a time of $T = 4$, using a time-step size of $\Delta t = 0.2$. For this experiment, we generated trajectories within the energy range $[-4, 4]$.

After learning a suitable quadratic embedding with the given set-up in Appendix A, we plot the training data of the Lotka-Volterra equations in phase space in Figure 5.4a. In this example, we focus on trajectories in phase space that do not complete a full orbit of the energy level. Furthermore, we demonstrate the learned dynamics in the phase space for a random initial value in Figure 5.4b for the Lotka-Volterra equations. The figure shows that the learned model is accurate even in terms of predicting the orbit level of random test initial conditions.

In Figure 5.5, we compare time-domain simulations between the learned and ground truth models for the Lotka-Volterra equations, along with the corresponding absolute error. The simulations were conducted using a random initial condition, distinct from the initial trajectories used in the training set. The results depicted in Figure 5.5 demonstrate a high level of agreement between the dynamics of the Lotka-Volterra equations and the ground truth model, even after the final training time $T = 4$. In

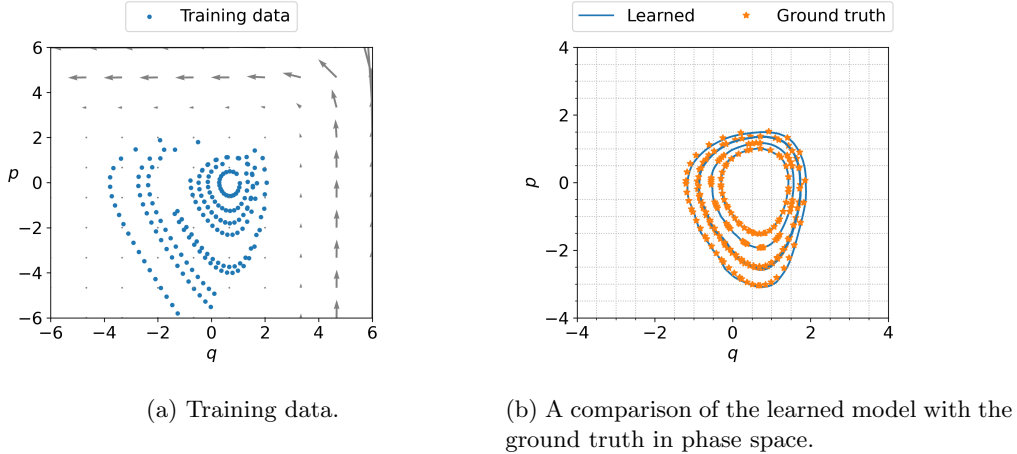


Figure 5.4: Lotka–Volterra: plot (a) shows training data in phase space, and plot (b) shows a comparison of the learned model with the ground truth in phase space with five random initial test conditions.

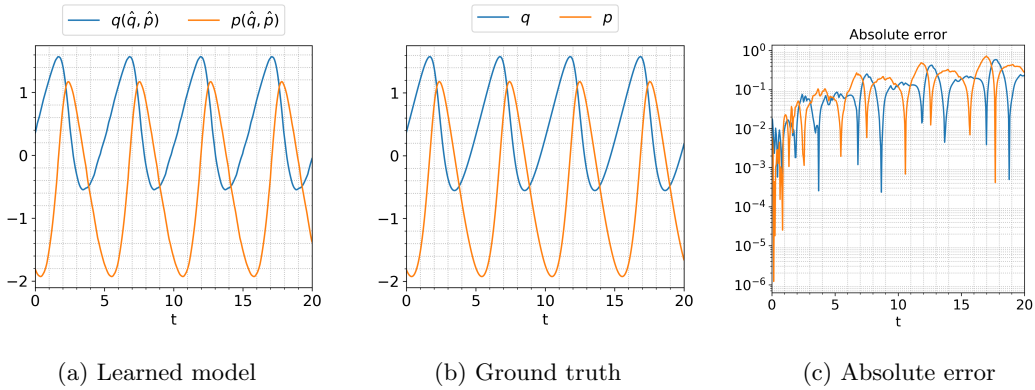


Figure 5.5: Lotka–Volterra: A comparison of time-domain simulation obtained using the learned model with the ground truth model for the Lotka–Volterra equations and using a random test initial condition.

Figure 5.6, we present the learned and canonical Hamiltonians for the Lotka–Volterra equation. Evidently, all Hamiltonians remain constant over time with minor fluctuations. We observe that these fluctuations primarily arise from errors in the autoencoder component, as the symplecticity condition is (weakly) applied to the encoder part but not the decoder part. Additionally, the fluctuations in the Hamiltonian error could be attributed to the training data, which is constructed from trajectories with a shorter time span compared to the pendulum example. The constant offset of the Hamiltonians corresponds to a different choice of energy null-level in the original space and the latent space, respectively. Since the Hamiltonian is a relative quantity, the overall performance of the learned model is linked to the error plot between the ground truth Hamiltonian and the learned Hamiltonian in the original space, where the Figure 5.6 shows that they coincide.

5.1.3. Nonlinear Oscillator

Our last low-dimensional example is a nonlinear (an-harmonic) oscillator with Hamiltonian

$$\mathcal{H}(q, p) = \frac{p^2}{2} + \frac{q^2}{2} + \frac{q^4}{4}, \tag{5.3}$$

where the natural frequency and the mass of the oscillator are considered to be unity.

To learn dynamics from data, we initially generated 20 random trajectories which are simulated up to final time $T = 4$ with step-size $\Delta t = 0.14$. The generated trajectories are in the energy range of $[-1, 1]$ for this task. In Figure 5.7a, we plot the training data of the nonlinear oscillator in phase space.

Having learned the desired embeddings, we next present a comparison of the learned model over three random initial points in Figure 5.7b, where the model captures the dynamics of the learned model with

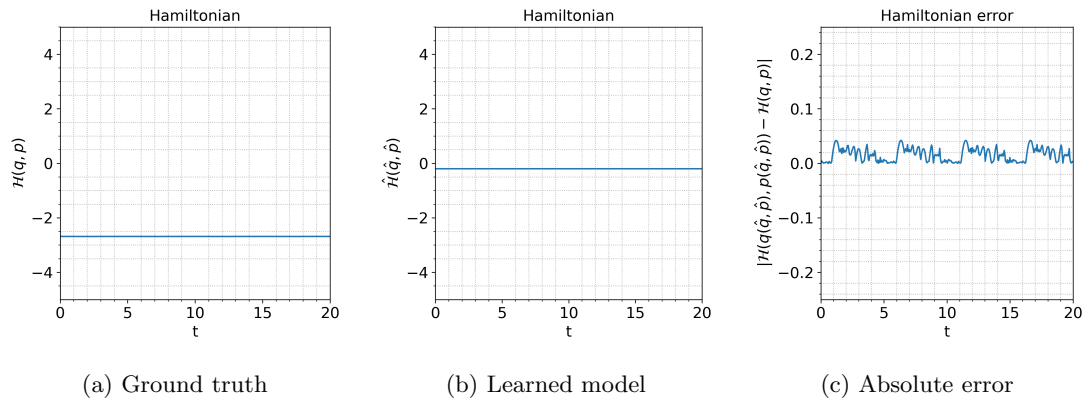


Figure 5.6: Lotka–Volterra: A comparison of the Hamiltonian in canonical coordinates for the ground truth model $\mathcal{H}(q, p)$, the learned Hamiltonian $\hat{\mathcal{H}}(\hat{q}, \hat{p})$ in the latent space, and the difference between the ground truth model and the learned model in the original space $\mathcal{H}(q(\hat{q}, \hat{p}), p(\hat{q}, \hat{p}))$ along time using a random test initial condition.

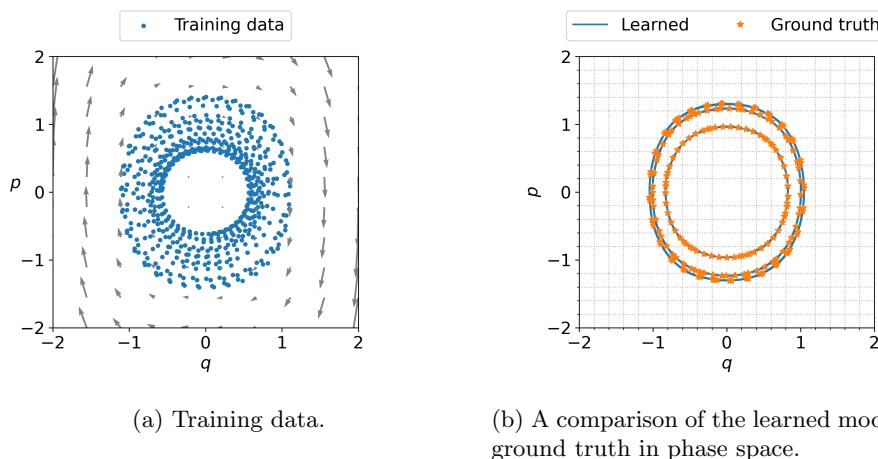


Figure 5.7: Nonlinear oscillator: Plot (a) shows training data in phase space, and Plot (b) shows a comparison of the learned model with the ground truth in phase space with three random initial test conditions.

good accuracy. In Figure 5.8, we demonstrate the temporal evolution of the learned model, the ground truth model for a nonlinear oscillator, and the corresponding absolute error in the time domain for a randomly chosen initial condition. The figure shows that dynamics are well captured over a long time horizon, exceeding the final training time $T = 4$. Furthermore, in Figure 5.9, we plot the learned and canonical Hamiltonians for the nonlinear oscillator, which demonstrates that all Hamiltonians remain constant over time with minor fluctuations, as seen in previous examples.

5.2. High-dimensional Systems

Next, we focus on learning low-dimensional models for high-dimensional data coming from high-dimensional systems.

5.2.1. Linear Wave Equation

We begin by considering a simple linear wave equation of the form:

$$\begin{aligned} u_{tt} &= cu_{xx}, \\ u(t_0, x) &= u^0(x), \quad x \text{ in } \Omega, \end{aligned} \tag{5.4}$$

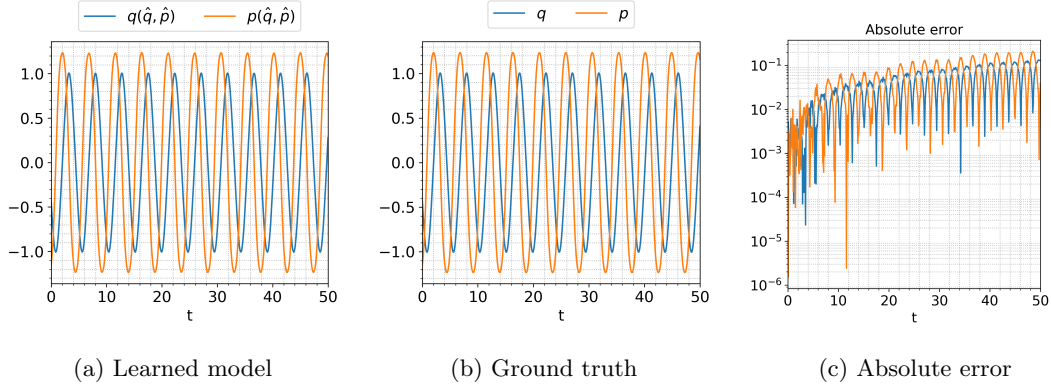


Figure 5.8: Nonlinear oscillator: Comparison of the learned model with the ground truth model for the harmonic oscillator in the time axis using a random initial condition.

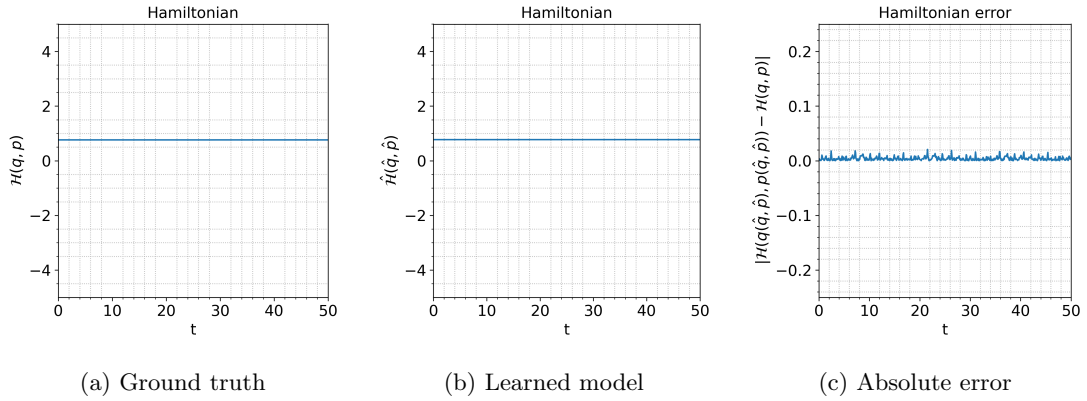


Figure 5.9: Nonlinear oscillator: A comparison of the Hamiltonian in canonical coordinates for the ground truth model $\mathcal{H}(q, p)$, the learned Hamiltonian $\hat{\mathcal{H}}(\hat{q}, \hat{p})$ in the latent space, and the difference between the ground truth model and the learned model in the original space $\mathcal{H}(q(\hat{q}, \hat{p}), p(\hat{q}, \hat{p}))$ along time using a random test initial condition.

where c is the transport velocity, and boundary conditions are set to be periodic. The wave equation is an example of a Hamiltonian PDE. By defining the variables $p = u_t$ and $q = u$, we can obtain the Hamiltonian form of the wave equation [30], which is given by

$$\frac{\partial z}{\partial t} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \nabla_z \mathcal{H}, \quad z = \begin{bmatrix} q \\ p \end{bmatrix}, \quad (5.5)$$

where the Hamiltonian is given as

$$\mathcal{H}(u) = \frac{1}{2} \int_{\Omega} cq_x^2 + p^2 dx.$$

Next, we discretize the Hamiltonian form of the wave equation and obtain the following semi-discrete Hamiltonian ODE systems:

$$\frac{dz}{dt} = \mathbf{K}z, \quad (5.6)$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 0_N & I_N \\ cD_{xx} & 0_N \end{bmatrix},$$

$D_{xx} \in \mathbb{R}^{N \times N}$ is the three-point central difference approximation of ∂_{xx} , $0_N \in \mathbb{R}^{N \times N}$ is a matrix of zeros, $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix, and (\mathbf{q}, \mathbf{p}) are the discretized (q, p) .

In this task, we focus on learning a single wave equation over a single trajectory. For this purpose, we set the initial condition to $u^0(x) = \text{sech}(x)$. For training purposes, we have generated data of the wave equation on the domain $\Omega = [-5, 5]$ up to time $T = 20$ with time-step size $\Delta t = 0.05$. We set the spatial

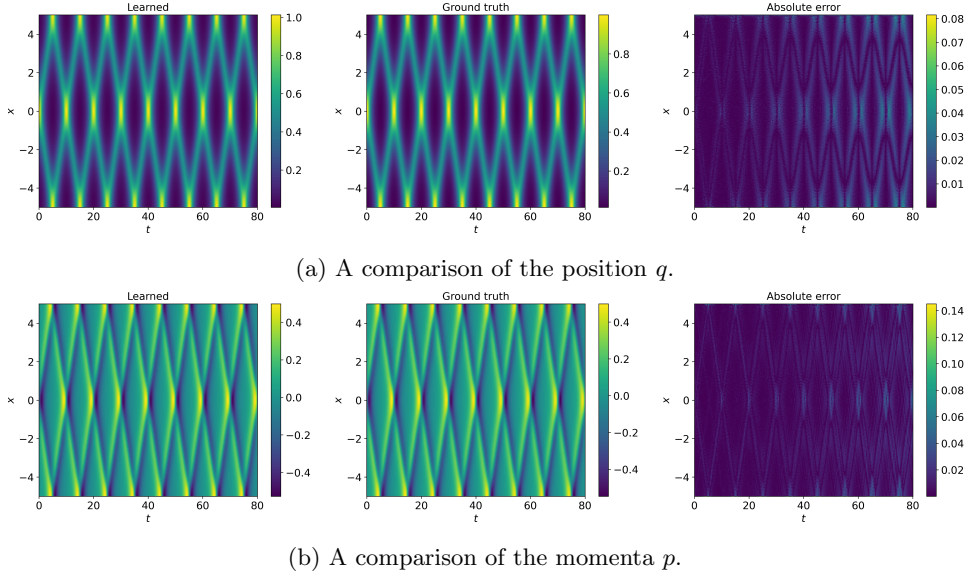


Figure 5.10: Wave equation: Comparisons of the position q and momenta p obtained using the learned model with the ground truth wave model (5.5).

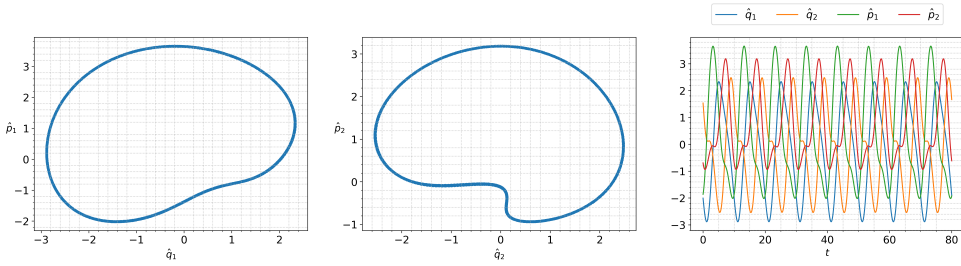


Figure 5.11: Wave equation: Learned variables in phase space and time domain.

dimension for the ground truth model of wave equation (5.6) to $2N = 1024$ and the learned problem dimension to $2n = 4$.

We compare the learned model with the ground truth in Figures 5.10a and 5.10b, for the states q and p , respectively. The figures show that the obtained model is stable and accurate over a long time horizon. Next, we examine the learned variables in phase space and time domain in Figure 5.11, which shows that the learned variables are orbiting on one particular energy level in phase space and are stable in the time domain as well.

5.2.2. Nonlinear Schrödinger Equation

Finally, we test the ability of our model to learn the nonlinear Schrödinger (NLS) equation in the last example of a high-dimensional problem. The NLS equation has various use cases, e.g., small-amplitude gravity waves on the surface of deep water with zero-viscosity, in the study of Bose-Einstein condensation, and the propagation of light in nonlinear optical fibers. Specifically, we look at the cubic Schrödinger equation which is given [31] by

$$\begin{aligned}
 i \frac{\partial u}{\partial t} + \alpha u_{xx} + \beta |u|^2 u &= 0, \\
 u(t_0, x) &= u^0(x), \quad x \text{ in } \Omega,
 \end{aligned}
 \tag{5.7}$$

with periodic boundary conditions. In the NLS equation (5.7), the parameter α is a non-negative constant and the constant parameter β is the focusing—with negative—and defocusing—with positive—values. In this example, we have fixed the parameters to $\alpha = \frac{1}{2}$, $\beta = 1$, and the domain is fixed to $\Omega = [-10, 10]$.

To obtain the canonical Hamiltonian form of the NLS equation (5.7), we write the complex-valued solution u in terms of its imaginary and real parts as $u = q + ip$. Then, the Hamiltonian of the NLS

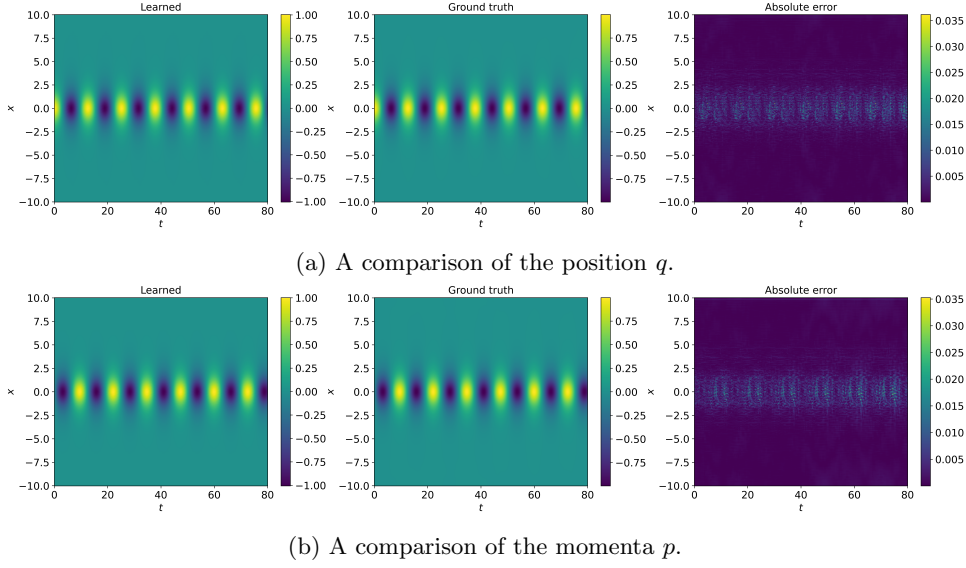


Figure 5.12: Nonlinear Schrödinger equation: Comparisons of the position q and momenta p obtained using the learned model with the ground truth wave model (5.7).

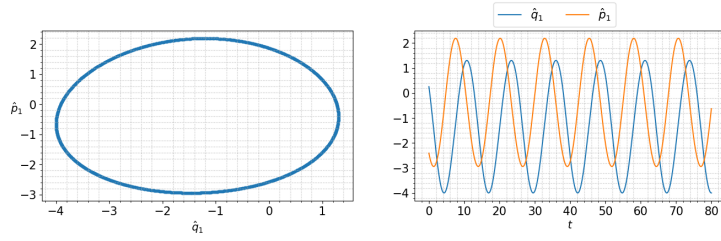


Figure 5.13: Nonlinear Schrödinger equation: Learned variables in phase space and time domain.

equations reads as

$$\mathcal{H}(u) = \frac{1}{2} \int_{\Omega} \left[\alpha \left(\frac{\partial q}{\partial x} \right)^2 + \alpha \left(\frac{\partial p}{\partial x} \right)^2 - \frac{\beta}{2} (q^2 + p^2)^2 \right] dx.$$

We used the same discretization as in the linear wave equation. The NLS equation was simulated with the initial condition $u^0(x) = \text{sech}(x)$ in the time domain up to final time $T = 80$ with time-step size $\Delta t = 0.05$. The spatial dimension for the ground truth model was fixed to $2N = 1024$. We used half of the obtained data, i.e., up to $T_{\text{train}} = 40$, to train our model. We set the dimension of the learned model to $2n = 2$.

Having learned the desired embedding, in Figures 5.12a and 5.12b, we present a comparison of the dynamics of the ground truth model (5.7) and the learned model, as well as the corresponding absolute error, in the time domain for the states q and p , respectively. The figures show that the learned model agrees with the ground truth model to a high degree of accuracy and can infer the dynamics of the nonlinear Schrödinger equation (5.7).

Lastly, we plot the dynamics of the learned phase space and time domain simulation for the ground truth model (5.7) in Figure 5.13 to present the stability of the learned dynamics. Figure 5.13 shows that the learned model is suitable for long time integration.

6. Conclusions

In this work, we have discussed the concept of data-driven quadratic symplectic representations of nonlinear Hamiltonian systems. We have defined an embedding as the lifting of original data coming from nonlinear Hamiltonian systems using a symplectic transformation, resulting in quadratic systems that describe the dynamics in the lifted space, with a cubic function as the Hamiltonian. The symplectic structure of the dynamics can be enforced by using symplectic auto-encoders and symmetric tensors.

This approach enables us to obtain a learned symplectic lifting. Additionally, for high-dimensional data, we discuss symplectic reduction to achieve a quadratic representation, leading to a low-dimensional quadratic Hamiltonian system. The advantage of this approach over structure-preserving model order reduction is that we directly learn the reduced dynamics fitting the data, eliminating the need for hyper reduction methods or taking gradients through the auto-encoder. We note that the proposed methodology does not require to know the full-order model in a discretized form.

We have demonstrated the efficiency of the proposed methodology by means of several low-dimensional and high-dimensional examples, illustrating the preservation of the Hamiltonian, i.e., energy, and long-term stability in extrapolation settings. In our future work, we investigate the effect of noise on the performance of the methodology and propose suitable treatments to it, for example, tailoring the approach proposed in [32]. Additionally, extensions to discrete Hamiltonian systems, and parametric and externally controlled Hamiltonian systems would be valuable contributions.

A. Implementation Details

Tables A.1 and A.2 contain all the necessary hyper-parameters for our illustrative examples. We set the hyper-parameters experimentally by monitoring the performance of the learned model on training data. For the symplectic lifting case, we have set the hyper-parameters $(\lambda_1, \lambda_2, \lambda_3)$ to $(10^{-1}, 1, 1)$ by monitoring all the losses to obtain a balanced decrease of all the losses simultaneously, while in the symplectic reduction case, the hyper-parameters $(\lambda_1, \lambda_2, \lambda_3)$ are set to $(1, 10^{-1}, 10^{-1})$ for the same goal. In order to deal with the inaccuracy of the reconstruction due to the structure of the auto-encoder, we have applied the penalisation:

$$\mathcal{L}_{\text{Rec}} = 0.5 \|x(t) - \phi(\psi(x(t)))\|_1, \quad (\text{A.1})$$

where $\|\cdot\|_1$ denotes the mean absolute error, averaged over all samples and dimensions. Similarly, we have penalized the parameters of $\mathcal{L}_{\dot{z}\dot{x}}$ with the mean absolute error scaled with a hyper-parameter 10^{-5} . Finally, we used fixed decay in both the symplectic reduction and lifting cases, using the `StepLR` implementation in `PyTorch`. We experimentally fixed both the decay rate and the decay step by monitoring the decay of the total loss function.

For the symplectic lifting case, we have used a Multi Layer Perceptron (MLP) architecture with skip connections and three hidden layer.

Parameters	Pendulum example	Lotka-Volterra example	Nonlinear oscillator example
Encoder layers [neurons]	[64, 64, 64]	[32, 32, 32]	[32, 32, 32]
Lifted coordinate system dimension	4	4	4
Learning rate	$3 \cdot 10^{-3}$	$3 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
Batch size	5	5	20
Activation function	<code>selu</code>	<code>selu</code>	<code>selu</code>
Weight decay	10^{-5}	10^{-5}	10^{-5}
Epochs	5501	4501	3501
Tolerance	$5 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$5 \cdot 10^{-2}$

Table A.1: The table contains all the hyper-parameters to learn the dynamics of the low-dimensional examples.

For the symplectic reduction case, we used a similar deep convolutional network (DCA) structure as the one given in [18]. In Figure A.1, we give the details of the auto-encoder structure.

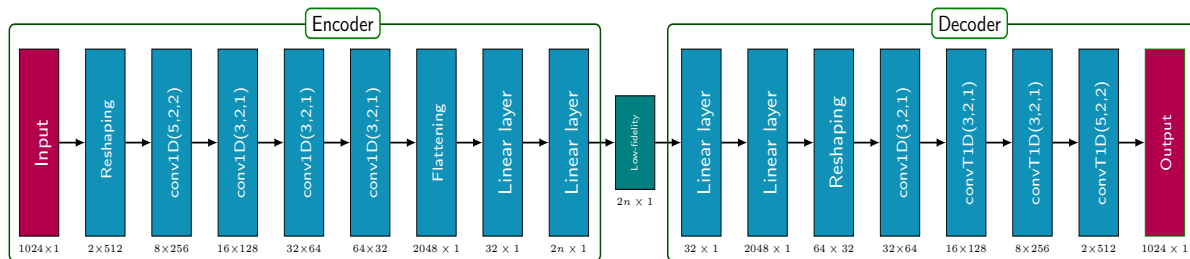


Figure A.1: The figure summarises the encoder and decoder architectures. $\text{conv1D}(k, s, p)$ denotes a 1D convolution layer with kernel size k , stride size s , padding size p and similarly, $\text{convT1D}(k, s, p)$ is a 1D transpose convolution layer with transpose kernel size k , stride size s , padding size p . We have used output padding size 1 to obtain symmetric auto-encoder structure in 1D transpose convolution layers. We denote the size of the output block below each block.

Parameters	Wave example	NLS example
Lifted coordinate system dimension	4	2
Learning rate	10^{-3}	10^{-3}
Batch size	50	50
Activation function	selu	selu
Weight decay	10^{-5}	10^{-5}
Epochs	6001	6001
Tolerance	$5 \cdot 10^{-2}$	$1 \cdot 10^{-2}$

Table A.2: The table contains all the hyper-parameters to learn the dynamics of the high-dimensional examples.

Funding Statement

Süleyman Yıldız and Peter Benner are partially supported by the German Research Foundation (DFG) Research Training Group 2297 “MathCoRe”, Magdeburg.

References

- [1] V. I. Arnol’d, *Mathematical methods of classical mechanics*. Springer New York, 1989.
- [2] C. L. Siegel and J. K. Moser, *Lectures on celestial mechanics*. Springer Berlin, Heidelberg, 1995.
- [3] R. Salmon, “Hamiltonian fluid mechanics,” *Annu. Rev. Fluid Mech.*, vol. 20, no. 1, pp. 225–256, 1988.
- [4] E. Faou, *Geometric numerical integration and Schrödinger equations*, vol. 15. European Mathematical Society, 2012.
- [5] J. P. Crutchfield, “Between order and chaos,” *Nat. Phys.*, vol. 8, no. 1, pp. 17–24, 2012.
- [6] B. Lusch, J. N. Kutz, and S. L. Brunton, “Deep learning for universal linear embeddings of nonlinear dynamics,” *Nat. Commun.*, vol. 9, no. 1, p. 4950, 2018.
- [7] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks,” *Proc. Math. Phys. Eng. Sci.*, vol. 474, no. 2213, p. 20170844, 2018.

- [8] R. Fang, D. Sondak, P. Protopapas, and S. Succi, “Neural network models for the anisotropic Reynolds stress tensor in turbulent channel flow,” *J. Turbul.*, vol. 21, no. 9-10, pp. 525–543, 2020.
- [9] K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annu. Rev. Fluid Mech.*, vol. 51, pp. 357–377, 2019.
- [10] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNs for time series classification,” *Neural Netw.*, vol. 116, pp. 237–245, 2019.
- [11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Inferring solutions of differential equations using noisy multi-fidelity data,” *J. Comput. Phys.*, vol. 335, pp. 736–746, 2017.
- [12] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Data-driven discovery of partial differential equations,” *Sci. Adv.*, vol. 3, no. 4, p. e1602614, 2017.
- [13] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, “Symplectic recurrent neural networks,” *arXiv preprint arXiv:1909.13334*, 2019.
- [14] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [15] M. Finzi, K. A. Wang, and A. G. Wilson, “Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 13880–13889, Curran Associates, Inc., 2020.
- [16] C. Offen and S. Ober-Blöbaum, “Symplectic integration of learned Hamiltonian systems,” *Chaos*, vol. 32, no. 1, p. 013122, 2022.
- [17] P. Goyal and P. Benner, “Generalized quadratic-embeddings for nonlinear dynamics using deep learning,” *arXiv preprint arXiv:2211.00357*, 2022.
- [18] P. Buchfink, S. Glas, and B. Haasdonk, “Symplectic model reduction of Hamiltonian systems on non-linear manifolds and approximation with weakly symplectic autoencoder,” *SIAM J. Sci. Comput.*, vol. 45, no. 2, pp. A289–A311, 2023.
- [19] L. Peng and K. Mohseni, “Symplectic model reduction of Hamiltonian systems,” *SIAM J. Sci. Comput.*, vol. 38, no. 1, pp. A1–A27, 2016.
- [20] B. Maboudi Afkham and J. S. Hesthaven, “Structure preserving model reduction of parametric Hamiltonian systems,” *SIAM J. Sci. Comput.*, vol. 39, no. 6, pp. A2616–A2644, 2017.
- [21] H. Sharma, H. Mu, P. Buchfink, R. Geelen, S. Glas, and B. Kramer, “Symplectic model reduction of Hamiltonian systems using data-driven quadratic manifolds,” *arXiv preprint arXiv:2305.15490*, 2023.
- [22] J. M. Lee, *Introduction to Smooth Manifolds*. Springer, 2012.
- [23] M. A. Savageau and E. O. Voit, “Recasting nonlinear differential equations as S-systems: a canonical nonlinear form,” *Math. Biosci.*, vol. 87, no. 1, pp. 83–115, 1987.
- [24] C. Gu, “QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems,” *IEEE Trans. Comput. Aided Des. Integr. Circuits. Syst.*, vol. 30, no. 9, pp. 1307–1320, 2011.
- [25] E. Qian, B. Krämer, B. Peherstorfer, and K. Willcox, “Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems,” *Physica D*, vol. 406, no. 1, p. art. 132401, 2020.
- [26] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, “Hamiltonian neural networks for solving equations of motion,” *Phys. Rev. E*, vol. 105, no. 6, p. 065305, 2022.
- [27] D. Strauch, *Classical Mechanics: An Introduction*. Springer Berlin Heidelberg, 2009.

- [28] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, “Symmetric tensors and symmetric tensor rank,” *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1254–1279, 2008.
- [29] Y. Tong, S. Xiong, X. He, G. Pan, and B. Zhu, “Symplectic neural networks in Taylor series form for Hamiltonian systems,” *J. Comput. Phys.*, vol. 437, p. 110325, 2021.
- [30] T. J. Bridges and S. Reich, “Numerical methods for Hamiltonian PDEs,” *J. Phys. A Math. Theor.*, vol. 39, no. 19, p. 5287, 2006.
- [31] B. Karasözen and M. Uzunca, “Energy preserving model order reduction of the nonlinear Schrödinger equation,” *Adv. Comput. Math.*, vol. 44, no. 6, pp. 1769–1796, 2018.
- [32] P. Goyal and P. Benner, “Neural ordinary differential equations with irregular and noisy data,” *Roy. Soc. Open Sci.*, vol. 10, no. 7, p. 221475, 2023.