

# Deep Learning for Structure-Preserving Universal Stable Koopman-Inspired Embeddings for Nonlinear Canonical Hamiltonian Dynamics

Pawan Goyal\*\* Süleyman Yıldız\* Peter Benner†‡

\*\*Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.

Email: [goyalp@mpi-magdeburg.mpg.de](mailto:goyalp@mpi-magdeburg.mpg.de), ORCID: [0000-0003-3072-7780](https://orcid.org/0000-0003-3072-7780)

\*Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.

Email: [yildiz@mpi-magdeburg.mpg.de](mailto:yildiz@mpi-magdeburg.mpg.de), ORCID: [0000-0001-7904-605X](https://orcid.org/0000-0001-7904-605X)

†‡Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany.

Email: [benner@mpi-magdeburg.mpg.de](mailto:benner@mpi-magdeburg.mpg.de), ORCID: [0000-0003-3362-4103](https://orcid.org/0000-0003-3362-4103)

††Otto von Guericke University, Universitätsplatz 2, 39106 Magdeburg, Germany

Email: [peter.benner@ovgu.de](mailto:peter.benner@ovgu.de)

## Abstract:

Discovering a suitable coordinate transformation for nonlinear systems enables the construction of simpler models, facilitating prediction, control, and optimization for complex nonlinear systems. To that end, Koopman operator theory offers a framework for global linearization for nonlinear systems, thereby allowing the usage of linear tools for design studies. In this work, we focus on the identification of global linearized embeddings for canonical nonlinear Hamiltonian systems through a symplectic transformation. While this task is often challenging, we leverage the power of deep learning to discover the desired embeddings. Furthermore, to overcome the shortcomings of Koopman operators for systems with continuous spectra, we apply the lifting principle and learn global cubicized embeddings. Additionally, a key emphasis is paid to enforce the bounded stability for the dynamics of the discovered embeddings. We demonstrate the capabilities of deep learning in acquiring compact symplectic coordinate transformation and the corresponding simple dynamical models, fostering data-driven learning of nonlinear canonical Hamiltonian systems, even those with continuous spectra.

**Keywords:** Canonical Hamiltonian systems, nonlinear systems, linear systems, symplectic transformation, Koopman operator, lifting-principle, continuous spectra.

## Novelty statement:

- Inspired by the Koopman operator theory, in this work, we address a data-driven discovery of global linearized embeddings for canonical Hamiltonian systems via a symplectic transformation.
- To address the limitation of Koopman operator theory for the systems with continuous spectra, we propose the discovery of global cubicized embeddings, influenced by the lifting principle.
- An essential aspect of our study involves enforcing the dynamic stability of the learned embeddings.
- We harness the power of deep learning, particularly auto-encoders, to effectively learn the desired embeddings.
- To demonstrate the effectiveness of our proposed methodologies, we provide several examples, including those in high-dimension, to show their applications and capabilities.

## 1. Introduction

Hamiltonian mechanics provides a potent mathematical framework for describing the intricate behaviors of diverse physics phenomena across a spectrum of scientific disciplines. This framework finds its applications in physics, engineering, and chemistry. Noteworthy applications span from the realms of solid-state physics, mechanical structures, and robotics to celestial mechanics, climate modeling, and beyond. Hamiltonian systems are characterized by their ability to conserve fundamental physical quantities, such as energy, momentum, symplecticity, and symmetries. As a result, it makes them a natural framework for representing a diverse range of physical phenomena [1–3]. Leveraging these inherent properties during the course of engineering studies (e.g., control, prediction, and optimization) and modeling enables us to attain our objectives effectively.

To capture complex nonlinear dynamics, nonlinear modeling is a key tool. While physics-based nonlinear dynamic modeling has historically been employed to comprehend complex processes, it is a formidable challenge to describe the underlying dynamics completely and accurately. With the surge in sensor technology, data availability has been more than ever. This, in turn, has intensified the pursuit of developing data-driven dynamic models that capture the underlying nonlinear behavior.

Learning nonlinear models using data has remarkably progressed in recent years—from classical techniques to neural networks, see, e.g., [4–12]. However, the quest for a comprehensive framework for nonlinear system modeling remains ongoing, particularly, towards utilizing physics-knowledge at an abstract level to enhance interpretability and generalizability. In this work, we seek to explore the characteristics of Hamiltonian dynamics. To that end, we begin by highlighting the Hamiltonian neural network (HNN) framework [13], in which the central idea is energy-based modeling. That is, instead of modeling the vector field with neural networks, the underlying Hamiltonian is learned using a neural network, and then the dynamical model is constructed so that it is symplectic. HNN has inspired several studies to generalize the framework to dissipative systems [14], control systems [15], port-Hamiltonian systems [16, 17], and Poisson systems [18]. Moreover, the operator inference framework [10] has been extended to Hamiltonian systems [19–22] but they require a prior model hypothesis. However, engineering design studies using nonlinear models are challenging. Therefore, encapsulating nonlinear dynamics within a weakly nonlinear framework, or even within a linear framework, is appealing since it offers us to employ tools for weakly nonlinear systems.

In this direction, Koopman operator theory [23, 24] provides a methodology, indicating that nonlinear systems can be globally linearized in suitable coordinate systems or embeddings, and it can be of an infinite dimensional. However, dealing with infinite-dimensional space, though conceptually intriguing, poses challenges, thus motivating us to obtain a finite-dimensional approximation. One popular technique is the dynamic mode decomposition (DMD) [25–27], which provides a means to achieve this goal. Despite its popularity, it may fall short in capturing highly nonlinear dynamics as it relies on linear measurements. To further enrich measurements, Extended DMD (eDMD) [28] has been developed by defining hand-design nonlinear observers and using them to learn a linear operator through the classical DMD approach. A parallel idea can be found in the variational approach to conformation dynamics [29–31], which also enriches models with nonlinear observers, akin to the spirit of eDMD. Nevertheless, there may still be closure-related issues with regard to the Koopman invariant subspace [32]. Furthermore, these observers can be high-dimensional, which can be tackled using kernel methods [33]. However, this might lead to an un-interpretable outcome or present challenges in reconstructing the relevant quantity of the interest. Moreover, these above-mentioned methodologies do not explicitly make use of the properties related to canonical Hamiltonian dynamics present in the data; hence, the globally linearized models are prone to not adhering to the physical properties of Hamiltonian dynamics.

In this work, we focus on identifying global linearized Koopman embeddings for canonical Hamiltonian dynamics. For this, we harness the capabilities of deep neural networks (DNNs) as a powerful computational tool. In the past, DNNs have been used to learn Koopman embeddings for nonlinear systems, see, e.g., [34–40] but without making use of Hamiltonian properties. Recently, the work [41] discusses learning Koopman embeddings while focusing on energy preservation. However, it does not use the symplectic property of canonical Hamiltonian systems, and thus, it does not guarantee that the auto-encoder defining the embeddings is a symplectic auto-encoder, e.g., [42, 43]. Furthermore, it does not guarantee bounded stability for the dynamics of the embeddings. To address this, we propose identifying Koopman-inspired global linearized embedding for canonical Hamiltonian systems so that the embedding dynamics are also canonical, and a symplectic nonlinear transformation yields embeddings. Additionally, we discuss how to enforce the bounded stability for the embeddings, which is arguably the most critical property related to dynamical systems.

Furthermore, for nonlinear systems, exhibiting continuous spectra of eigenvalues, obtaining finite-dimensional Koopman embeddings is rather difficult. To deal with it, the work [40] proposed the use of an auxiliary neural network designed to model the eigenvalues of linearized operators. In this case, the dynamics for the embeddings are not purely linear, as the operator depends on the output of the neural network, which is actually nonlinear. To alleviate the dependency on the neural network, we alternatively deal with the systems of continuous spectra by constructing weakly nonlinear systems. It relies on the framework of the lifting principle. It guides us that smooth nonlinear systems, including continuous spectra ones, can be exactly written as polynomial systems with a finite-dimensional embedding. Moreover, polynomial systems can exactly be written as finite-dimensional quadratic or cubic polynomial systems, see, e.g., [44, 45]. These principles to learn quadratic systems or embeddings have been used to learn nonlinear dynamics systems, see [46]. Further, learning a suitable quadratic embedding using auto-encoders has been discussed in [47], which has been extended to canonical Hamiltonian systems while carefully treating the properties of the Hamiltonian dynamics [43]. However, a major drawback of learning quadratic embeddings in the context of canonical Hamiltonian systems is the preservation of bounded stability. To that end, we discuss a global cubicized framework for canonical systems that not only allows the construction of stable dynamics for embeddings but aims to mitigate the problem with continuous spectra in the classical Koopman framework as well. By embracing this framework, we seek to enhance the understanding and applicability of canonical Hamiltonian systems within a broader context.

The remaining paper is organized as follows. In Section 2, we recall the definition of canonical Hamiltonian systems and the property of a symplectic transformation that is used to enforce the auto-encoder to be symplectic. In Section 3, we discuss a Koopman-inspired global linearized framework for canonical Hamiltonian systems while preserving their properties in the Koopman embeddings. We also emphasize how to ensure bounded stability for the dynamics of the embeddings. Moreover, we address the continuous spectra by means of learning weakly nonlinear systems—that is, a cubic nonlinear system with the Hamiltonian being a quartic function and discuss their stability as well. Section 4 presents numerical experiments, illustrating the performance of the proposed methodologies and presenting a comparison with [43]. Additionally, in Section 5, we discuss how to efficiently employ the proposed methodologies to high-dimensional data and present a comparison with [21, 22]. In Section 6, we conclude the paper with a summary and avenues for future research.

## 2. Symplectic Transformation

Our interest lies in learning canonical Hamiltonian systems of the form:

$$\dot{\mathbf{x}}(t) = \mathbf{J}_{2n} \nabla_{\mathbf{x}} \mathbf{H}(\mathbf{x}(t)) \in \mathbb{R}^{2n}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (2.1)$$

where the state  $\mathbf{x} \in \mathbb{R}^{2n}$  contains generalized position  $\mathbf{q} \in \mathbb{R}^n$  and generalized momenta  $\mathbf{p} \in \mathbb{R}^n$ , and  $\mathbf{J}_{2n}$  is a symplectic matrix, i.e.,  $\mathbf{J}_{2n} := \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$ , and  $\nabla_{\mathbf{x}}$  denotes the gradient with respect to  $\mathbf{x}$ . Moreover,  $\mathbf{x}_0 = (\mathbf{q}_0, \mathbf{p}_0) \in \mathbb{R}^{2n}$  denotes an initial condition. Furthermore, the Hamiltonian function  $\mathbf{H}: \mathbb{R}^{2n} \rightarrow \mathbb{R}$  defines the Hamiltonian of the system and is constant along the solution trajectories. Next, we recall a definition of a symplectic transformation from, e.g., [43].

**Definition 1** (e.g., [43]). *A map  $\psi$  is a symplectic transformation of  $2n$ -dimensional to  $2m$ -dimensional when the following condition is fulfilled:*

$$(\mathrm{d}\psi_{\mathbf{x}})^\top \mathbf{J}_{2m} \mathrm{d}\psi_{\mathbf{x}} = \mathbf{J}_{2n}, \quad \forall \mathbf{x} \in \mathbb{R}^{2n}, \quad (2.2)$$

where  $\mathrm{d}\psi_{\mathbf{x}} \in \mathbb{R}^{2m \times 2n}$  is the Jacobian of  $\psi$  with respect to  $\mathbf{x}$ . Furthermore, when  $m \geq 0$ , we refer to it as a symplectic lifting [43].

## 3. Data-Driven Canonical Hamiltonian Systems

Here, we set up our problem of learning canonical Hamiltonian systems using data. Specifically, in this work, we focus on continuous-time dynamical systems of the form:

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)), \quad (3.1)$$

where  $\mathbf{x}(t)$  is the state of the system, and the function  $\mathbf{f}$ —also referred to as vector field—defines its time evolution for a given initial condition and is often nonlinear. Additionally, we assume the state  $\mathbf{x}(t)$  to be low-dimensional; however, for high-dimensional  $\mathbf{x}$ , we can obtain a low-dimensional representation by leveraging coherent structures in the spatial space, which we shall discuss later in detail. Our primary objective is to learn the function  $\mathbf{f}$  from the given data  $\mathbf{x}$ , where  $\mathbf{f}$  can be arbitrarily complex and nonlinear.

In this work, we draw inspiration from the Koopman operator theory [23] and lifting principles [44, 45] to achieve our goal. In particular, we aim to learn the time evolution of  $\mathbf{x}$  through suitable observers, employing concepts from these theories to address our learning task.

### 3.1. Koopman-inspired linear symplectic representations and its stability

The Koopman theory offers a valuable approach to represent nonlinear dynamics as linear dynamical systems by means of the Koopman operators, thus allowing predictions and control for nonlinear systems by employing tools from linear theory. In particular, the Koopman theory indicates that it is possible to define a set of observers  $\mathbf{y}$  as a function of  $\mathbf{x}$ , such that the evolution of these observers can be described by a linear dynamical system, i.e.,

$$\frac{d}{dt}\mathbf{y} = \mathcal{K}\mathbf{y}, \quad (3.2)$$

where  $\mathcal{K}$  is a Koopman operator. However, determining those observers is a challenge. In this paper, these observers are referred to as embeddings.

In this work, we harness the capabilities of deep learning to identify finite-dimensional Koopman embeddings whose dynamics can be described by linear systems. In other words, we aim to approximate the infinite-dimensional Koopman operator by a finite-dimensional one. While several deep learning approaches have been explored in related contexts, none have specifically addressed canonical Hamiltonian systems. To that end, we aim to design embeddings  $\mathbf{y} = \phi(\mathbf{x})$ , where  $\phi$  can be regarded as construction of embeddings based on the measurement  $\mathbf{x}$  so that their dynamics can be given by a linear system of the form (2.1). Furthermore, it is crucial to enforce the requirement that the mapping of measurements to embeddings is a symplectic transformation. To accomplish these goals, the following objectives are pursued.

- **Learning linear representation.** Our goal is to learn  $2m$ -dimensional embeddings  $\mathbf{y}$  from the measurements  $\mathbf{x}$  using a mapping function  $\phi$ , i.e.,  $\mathbf{y} = \phi(\mathbf{x})$ , and  $\mathbf{x}$  can be recovered using  $\mathbf{x} = \phi^{-1}(\mathbf{y})$ . It is achieved by means of an autoencoder, where  $\phi$  acts as an encoder that takes  $\mathbf{x}$  to  $\mathbf{y}$ , and  $\psi := \phi^{-1}$  behaves like a decoder that takes  $\mathbf{y}$  back to  $\mathbf{x}$ . The quality of the autoencoder is evaluated by the loss as follows:

$$\mathcal{L}_{\text{encdec}} = \|\mathbf{x} - \psi(\phi(\mathbf{x}))\|. \quad (3.3)$$

- **Linear dynamics.** To ensure linear dynamics of the embeddings  $\mathbf{y}$ , we require that  $\mathbf{y}$  satisfies (2.1). Thus, there exists a Hamiltonian for  $\mathbf{y}$ , which takes a quadratic form, i.e.,  $\mathbf{H}^{(L)}(\mathbf{y}) = \mathbf{y}^\top \mathbf{A} \mathbf{y} + \mathbf{b}^\top \mathbf{y} + \mathbf{c}$ . As a result, we can write the dynamics of  $\mathbf{y}$  as follows:

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{J}_{2m} \nabla_{\mathbf{y}} \mathbf{H}^{(L)}(\mathbf{y}), \quad (3.4)$$

where  $\mathbf{J}_{2m}$  is a symplectic matrix, and  $\nabla_{\mathbf{y}} \mathbf{H}^{(L)}(\mathbf{y})$  represents the gradient of the Hamiltonian with respect to  $\mathbf{y}$ . This goal is assessed using the following objective function:

$$\mathcal{L}_{\text{deri}} = \left\| \nabla_{\mathbf{x}} \phi(\mathbf{x}) \dot{\mathbf{x}} - \mathbf{J}_{2m} \nabla_{\mathbf{y}} \mathbf{H}^{(L)}(\phi(\mathbf{x})) \right\|. \quad (3.5)$$

- **Symplectic transformation.** Finally, we note that the mapping  $\phi$  that takes  $\mathbf{x}$  to  $\mathbf{y}$  is required to be a symplectic transformation. Therefore, for  $m \geq n$ , we enforce it by using the following condition:

$$\mathcal{L}_{\text{symp}} = \left\| (d\phi_{\mathbf{x}})^\top \mathbf{J}_{2m} d\phi_{\mathbf{x}} - \mathbf{J}_{2n} \right\|. \quad (3.6)$$

We consider  $\|\cdot\|$  to be the mean-squared error. Using a weighted sum, we combine these three losses, given in (3.3), (3.5), and (3.6), thus resulting in the total loss as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{encdec}} + \lambda_2 \mathcal{L}_{\text{symp}} + \lambda_3 \mathcal{L}_{\text{deri}}, \quad (3.7)$$

where  $\lambda_{\{1,2,3\}}$  are hyper-parameters. Consequently, we can expect to learn the Koopman embeddings with symplectic properties from the given measurements  $\mathbf{x}$ .

**Stability guarantee.** Stability is a crucial property for differential equations. Thus, we next discuss the stability of the learned dynamical system for the identified embeddings  $\mathbf{y}$ . With this regard, we aim to study the stability of dynamical systems of the form:

$$\dot{\mathbf{y}}(t) = \mathbf{J}_{2m} \nabla_{\mathbf{y}} \mathbf{H}^{(L)}(\mathbf{y}), \quad (3.8)$$

where  $\mathbf{H}^{(L)}(\mathbf{y}) = \mathbf{y}^\top \mathbf{A} \mathbf{y} + \mathbf{y}^\top \mathbf{b} + \mathbf{c}$ . For the stability of the system (3.8), it is sufficient to ensure that  $\mathbf{H}^{(L)}(\mathbf{y})$  is bounded from below and radially unbounded. This means that

$$\mathbf{H}^{(L)}(\mathbf{y}) \geq \gamma, \quad \text{and} \quad \lim_{\|\mathbf{y}\| \rightarrow \infty} \mathbf{H}^{(L)}(\mathbf{y}) \rightarrow \infty, \quad \forall \mathbf{y} \in \mathbb{R}^{2m}, \quad (3.9)$$

where  $\gamma \in \mathbb{R}$  is a constant and finite. Since a constant in  $\mathbf{H}^{(L)}$  does not affect the dynamics for  $\mathbf{y}$ , we can assume  $\gamma$  to be zero without loss of generality. Next, we discuss a construction of Hamiltonian functions which are quadratic and satisfy (3.9).

**Theorem 1.** Consider a Hamiltonian  $\mathbf{H}^{(L)}(\mathbf{y}) = \mathbf{y}^\top \mathbf{A} \mathbf{y} + \mathbf{y}^\top \mathbf{b} + \mathbf{c}$  which can be written as

$$\mathbf{H}^{(L)}(\mathbf{y}) = \begin{bmatrix} \mathbf{y} \\ w \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{y} \\ w \end{bmatrix}, \quad (3.10)$$

where  $w$  is a scalar constant, and  $\mathbf{Q} = \mathbf{Q}^\top > 0$ . Then,

- (a)  $\mathbf{H}^{(L)}(\mathbf{y})$  is bounded from below, and
- (b)  $\mathbf{H}^{(L)}(\mathbf{y})$  is radially unbounded, i.e.,  $\lim_{\|\mathbf{y}\| \rightarrow \infty} \mathbf{H}^{(L)}(\mathbf{y}) \rightarrow \infty$ .
- (c) Furthermore, we have  $\|\mathbf{y}(t)\|_2^2 \leq \frac{\mathbf{H}^{(L)}(\mathbf{y}_0)}{\sigma_{\min}(\mathbf{Q})}$ ,  $\forall t \geq 0$ , where  $\mathbf{y}(t)$  is the solution of (3.8) at time  $t$  for a given initial condition  $\mathbf{y}_0$ , and  $\sigma_{\min}(\cdot)$  denotes the smallest singular values.

*Proof.* (a) Since  $\mathbf{H}^{(L)}(\mathbf{y})$  exhibits a sum-of-squares form, it is non-negative by construction. Hence, it is bounded from below, and precisely,  $\mathbf{H}^{(L)}(\mathbf{y}) \geq 0$ .

- (b) Using the symmetric positivity property of  $\mathbf{Q}$ , we have

$$\mathbf{H}^{(L)}(\mathbf{y}) \geq \sigma_{\min}(\mathbf{Q}) \left\| \begin{bmatrix} \mathbf{y}(t) \\ w \end{bmatrix} \right\|_2^2 \geq \sigma_{\min}(\mathbf{Q}) \|\mathbf{y}(t)\|_2^2.$$

Since  $\sigma_{\min}(\mathbf{Q}) > 0$ , it is clear that as  $\|\mathbf{y}(t)\|_2 \rightarrow \infty$ ,  $\mathbf{H}^{(L)}(\mathbf{y}) \rightarrow \infty$ . Hence,  $\mathbf{H}^{(L)}(\mathbf{y})$  is radially unbounded.

- (c) The dynamics of  $\mathbf{y}$  is given by a canonical Hamiltonian system of the form (3.8). For this, we know that the Hamiltonian is constant along the trajectory, meaning

$$\mathbf{H}^{(L)}(\mathbf{y}_0) = \mathbf{H}^{(L)}(\mathbf{y}(t)), \quad \forall t \geq 0,$$

where  $\mathbf{y}_0$  is an initial condition and  $\mathbf{y}(t)$  is the solution of (3.8) at time  $t$  for the given initial condition  $\mathbf{y}_0$ . Thus, we have

$$\begin{aligned} \mathbf{H}^{(L)}(\mathbf{y}_0) &= \mathbf{H}^{(L)}(\mathbf{y}(t)) \\ &= \begin{bmatrix} \mathbf{y}(t) \\ w \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{y}(t) \\ w \end{bmatrix} \geq \sigma_{\min}(\mathbf{Q}) \left\| \begin{bmatrix} \mathbf{y}(t) \\ w \end{bmatrix} \right\|_2^2 \geq \sigma_{\min}(\mathbf{Q}) \|\mathbf{y}(t)\|_2^2. \end{aligned}$$

Based on it, we immediately have the result (c), which concludes the proof.  $\square$

Utilizing the result of [Theorem 1](#), we can parameterize the Hamiltonian function for the observers  $\mathbf{y}$  to guarantee the stability of their time-evaluation.

### 3.2. Systems with continuous spectra and remedy by cubicized representations

Using the Koopman operator theory, one can strive to achieve a global linearization of nonlinear dynamical systems. However, in the case of certain nonlinear systems, such as the nonlinear pendulum, continuous spectra may be exhibited, making it difficult to learn a global linearization. Capturing the dynamics of such systems requires a high-dimensional embedding space to account for the continuous spectra. In a recent work [40], the eigenvalues of the linear Koopman operator are parameterized using a neural network that depends on the embeddings themselves, enabling richer dynamics to capture. However, it is then no longer a purely linear operator.

In this work, we instead take a different approach by exploring learning simpler weak nonlinear systems rather than linearized Koopman embeddings. It builds on the theory, articulating that smooth nonlinear dynamical systems can be adequately represented as polynomial systems using a finite number of observers [44, 45]. This is in contrast to the Koopman operator theory, which may require infinite-dimensional embeddings to represent nonlinear systems as linear, particularly those with continuous spectra. Furthermore, nonlinear polynomial systems can be written as quadratic or cubic systems with a few more, but still finite, observers. Based on these observations, the authors in [43] investigated a problem of learning representations for nonlinear canonical Hamiltonian systems with cubic polynomial Hamiltonian. However, it can be noticed that nonlinear canonical Hamiltonian systems with cubic Hamiltonian cannot be stable. This instability arises from the nature of the cubic polynomial functions, which are neither bounded from below nor from above. Consequently, even though the dynamics may exhibit the Hamiltonian conservation along the trajectories, the system can still be inherently unstable. An illustrative example is given in Example 1.

To address these drawbacks, namely with continuous spectra and Hamiltonian systems with cubic Hamiltonian, we seek to identify suitable embeddings from the measurements so that the dynamics of the embeddings can be given by a canonical Hamiltonian system of the form (2.1), where the Hamiltonian is a quartic polynomial function, instead of being cubic. Consequently, we expect to design finite-dimensional embeddings that can explain the dynamics of nonlinear Hamiltonian systems with continuous spectra while ensuring the dynamic stability of the embeddings. To achieve this, it is essential for the underlying Hamiltonian to be radially unbounded and non-negative. We delve into this topic in the following discussion.

**Theorem 2.** Consider a canonical Hamiltonian system of the form (2.1) as:

$$\dot{\mathbf{y}}(t) = \mathbf{J}_{2m} \nabla_{\mathbf{y}} \mathbf{H}^{(C)}(\mathbf{y}), \quad (3.11)$$

where  $\mathbf{H}^{(C)}(\mathbf{y})$  is a quartic Hamiltonian as follows:

$$\mathbf{H}^{(C)}(\mathbf{y}) = \mathbf{a}_0 + \mathbf{a}_1^\top \mathbf{y} + \mathbf{a}_2^\top (\mathbf{y} \otimes \mathbf{y}) + \mathbf{a}_3^\top (\mathbf{y} \otimes \mathbf{y} \otimes \mathbf{y}) + \mathbf{a}_4^\top (\mathbf{y} \otimes \mathbf{y} \otimes \mathbf{y} \otimes \mathbf{y}),$$

where  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$  are vectors of appropriate sizes and  $\otimes$  denotes the Kronecker product. If  $\mathbf{H}^{(C)}(\mathbf{y})$  can be written as a sum-of-square, i.e.,

$$\mathbf{H}^{(C)}(\mathbf{y}) = \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix}, \quad (3.12)$$

where  $w$  is a scalar constant and  $\mathbf{Q} = \mathbf{Q}^\top > 0$ . Then, the following hold:

1.  $\mathbf{H}^{(C)}(\mathbf{y})$  is bounded from below,
2.  $\mathbf{H}^{(C)}(\mathbf{y})$  is radially unbounded, i.e.,  $\lim_{\|\mathbf{y}\| \rightarrow \infty} \mathbf{H}^{(C)}(\mathbf{y}) \rightarrow \infty$ , and
3.  $\|\mathbf{y}(t)\|_2^2 + \|\mathbf{y}(t)\|_2^4 \leq \frac{\mathbf{H}^{(C)}(\mathbf{y}_0)}{\sigma_{\min}(\mathbf{Q})}$ , where  $\mathbf{y}(t)$  is the solution of (3.11) at time  $t$  for a given initial condition  $\mathbf{y}_0$ .

*Proof.* The proofs of (a) and (b) follow the same arguments given in Theorem 1; thus, we will omit them here. For (c), we note that

$$\mathbf{H}^{(C)}(\mathbf{y}_0) = \mathbf{H}^{(C)}(\mathbf{y}(t)), \quad \forall t \geq 0,$$



where  $\mathbf{y}(t_0)$  is an initial condition and  $\mathbf{y}(t)$  is the solution of (3.11) at time  $t$  for the given initial condition  $\mathbf{y}_0$ . Thus, we have

$$\begin{aligned} \mathbf{H}^{(C)}(\mathbf{y}_0) &= \mathbf{H}^{(C)}(\mathbf{y}(t)) \\ &= \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t) \otimes \mathbf{y}(t) \\ w \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t) \otimes \mathbf{y}(t) \\ w \end{bmatrix} \geq \sigma_{\min}(\mathbf{Q}) \left\| \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t) \otimes \mathbf{y}(t) \\ w \end{bmatrix} \right\|_2^2 \geq \sigma_{\min}(\mathbf{Q}) \left( \|\mathbf{y}(t)\|_2^2 + \|\mathbf{y}(t)\|_2^4 \right). \end{aligned}$$

In the above equation, we have used the relation  $\|\mathbf{y} \otimes \mathbf{y}\|_2^2 = \|\mathbf{y}\|_2^4$ . This completes the proof.  $\square$

The positive definite condition on the matrix  $\mathbf{Q}$  is sufficient for stability; in the following, we discuss a case where  $\mathbf{Q}$  can be semi-definite, yet the Hamiltonian system is stable.

**Lemma 1.** *Consider a canonical Hamiltonian system with a Hamiltonian*

$$\mathbf{H}^{(C)}(\mathbf{y}) = \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix}^\top \mathbf{Q} \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix}, \quad (3.13)$$

where  $\mathbf{y} \in \mathbb{R}^{2m}$  and  $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ . Let  $\mathbf{V} \in \mathbb{R}^{\hat{m}, \tilde{m}}$ , where  $\tilde{m} = 2m + 4m^2 + 1$  and  $\hat{m} < \tilde{m}$ , be a matrix so that

$$\mathbf{Q} = \mathbf{V}^\top \mathbf{Q}_1 \mathbf{V}, \quad (3.14)$$

where  $\mathbf{Q}_1 = \mathbf{Q}_1^\top > 0$ . If  $\left\| \mathbf{V} \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix} \right\|_2^2 \geq \mathbf{g}(\mathbf{y})$ , where the function  $\mathbf{g}(\mathbf{y})$  is radially unbounded, then the system is stable.

*Proof.* First, note that for stability, we require the underlying Hamiltonian to be bounded from below and radially unbounded. Since  $\mathbf{H}^{(C)}$  has a sum-of-squares form, it is bounded from below by construction. For radially unboundedness, we substitute for  $\mathbf{Q}$  from (3.14) in (3.13), yielding

$$\mathbf{H}^{(C)}(\mathbf{y}) = \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix}^\top \mathbf{V}^\top \mathbf{Q}_1 \mathbf{V} \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix} \geq \sigma_{\min}(\mathbf{Q}_1) \left\| \mathbf{V} \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \otimes \mathbf{y} \\ w \end{bmatrix} \right\|_2^2 = \sigma_{\min}(\mathbf{Q}_1) \mathbf{g}(\mathbf{y}). \quad (3.15)$$

Since the function  $\mathbf{g}$  is assumed to be radially unbounded, the Hamiltonian is also radially unbounded, thus concluding the proof.  $\square$

An illustrative example (see Example 2) demonstrates Lemma 1. Thus, we focus on the quartic hypothesis applied to the Hamiltonian, which can lead to the desired embedding- stable global cubicized embeddings for nonlinear canonical Hamiltonian dynamical systems. To achieve this, we follow similar principles discussed in the previous subsection, where we discussed learning linearized Koopman embeddings for nonlinear canonical Hamiltonian systems. However, the crucial distinction lies in the hypothesis about the Hamiltonian function, which now takes the form of a quartic function (as given in (3.12)), rather than a quadratic one.

**Remark 1.** *In Theorem 2 and Lemma 1, we utilized a general quartic function in the sum-of-squares form. However, we can relax it by using  $\mathbf{y} \circ \mathbf{y}$  instead of  $\mathbf{y} \otimes \mathbf{y}$ , where  $\circ$  denotes the Hadamard product. This modification allows us to reduce the number of parameters defining  $\mathbf{H}^{(C)}$  while still retaining a quartic function in a sum-of-squares form. Although this relaxation results in a slightly less general quartic function, we empirically observe in our numerical experiments that it is less prone to over-fitting; hence, we utilize this in our experiments.*

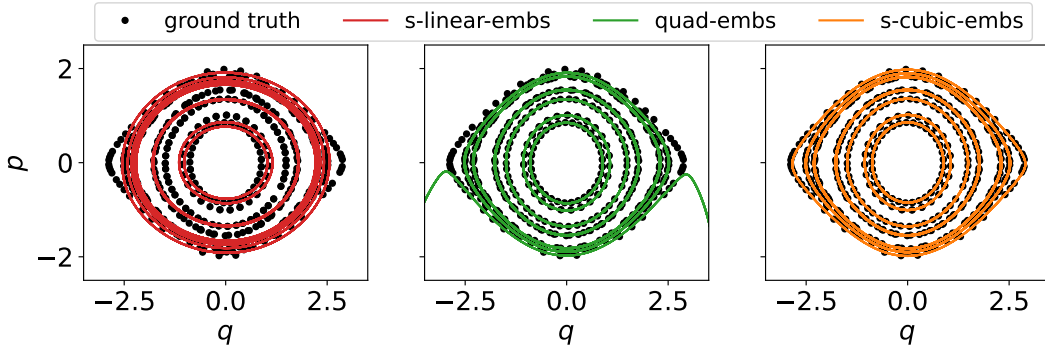


Figure 4.1: Nonlinear pendulum: The figure shows a comparison of various learned models with the ground truth in the phase space for test initial conditions.

## 4. Results for Low-dimensional Nonlinear Systems

In this section, we begin by examining the performance of the proposed methodologies for low-dimensional dynamical systems. We refer to the methods presented in Sections 3.1 and 3.2 as **s-linear-embs** and **s-cubic-embs**, respectively, where ‘s’ at the beginning indicates the stability guarantee. We consider three examples to evaluate their performance: the simple nonlinear pendulum, a harmonic oscillator, and the Lotka-Volterra equation. We compare our proposed methodologies with the one discussed in [43], which is also purely data-driven and aims to learn a quadratic representation for canonical Hamiltonian nonlinear systems. We refer to it as **quad-embs**. Furthermore, It is worth noting that all three methods considered involve auto-encoders based on neural networks. Therefore, we employ the same neural network architectures for all methods. Additionally, the training settings remain consistent across all three methods, which are discussed in detail in Appendix B. Lastly, we utilize the implicit midpoint rule as the time integrator that preserves the symplectic structure after time discretization; see [48].

### 4.1. Nonlinear pendulum

In the first example, we consider an ideal nonlinear pendulum in a normalized form, whose governing equations are given by

$$\begin{bmatrix} \dot{p}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} -\sin(q(t)) \\ p(t) \end{bmatrix}, \quad (4.1)$$

where  $q$  and  $p$  denote the position and momentum related to the pendulum dynamics. Moreover, the Hamiltonian of (4.1) is given by

$$\mathcal{H}(q, p) = (1 - \cos(q)) + \frac{1}{2}p^2, \quad (4.2)$$

To generate the training dataset, we follow [43]. The initial values of  $q$  and  $p$  are chosen within the range of  $[-3, 3] \times [-3, 3]$ ; however, to avoid a 360-degree swing of the pendulum, we choose them such that the energy  $\mathcal{H}(q, p) \leq 2$ . We consider 20 random initial conditions, and for each initial condition, we take 25 equidistant data points within the time interval  $[0, 20]$ .

Next, we learn the desired embeddings and their dynamics using **s-linear-embs**, **quad-embs**, and **s-cubic-embs**. To accomplish this, we fix the dimension of the embeddings to two and identify them by means of an auto-encoder for each of the three methods. The autoencoder architecture and training setup details can be found in Appendix B. After learning the desired embeddings, we evaluate the performance of the learned representation using 25 test initial conditions that were not part of the training set. In addition, for the testing phase, we consider a much longer time interval of  $[0, 50]$ , compared to the training time interval  $([0, 20])$ , and measure the performance using 2500 points within the testing interval.

First, we illustrate the quality of the learned  $q$  and  $p$  using the embeddings in phase space, shown in Figure 4.1. These figures clearly demonstrate that **s-linear-embs** fails to capture the dynamics accurately. This is potentially attributed to the fact that the nonlinear pendulum has continuous spectra, making it difficult to learn a finite-dimensional universal linear representation. Furthermore, **quad-embs** learns the dynamics well for small amplitudes  $q$  and  $p$ . However, it struggles to capture the dynamics for larger amplitudes, and its limitations become even more apparent when dealing with highly nonlinear



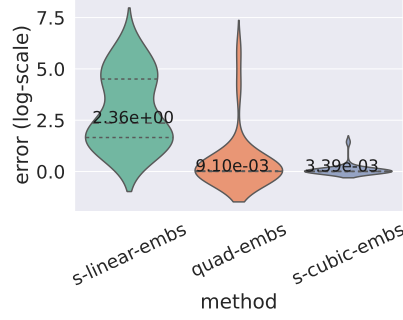


Figure 4.2: Nonlinear pendulum: A qualitative performance analysis of different methods for test initial conditions. The middle dashed line shows the median, with the precise number, across all test initial conditions.

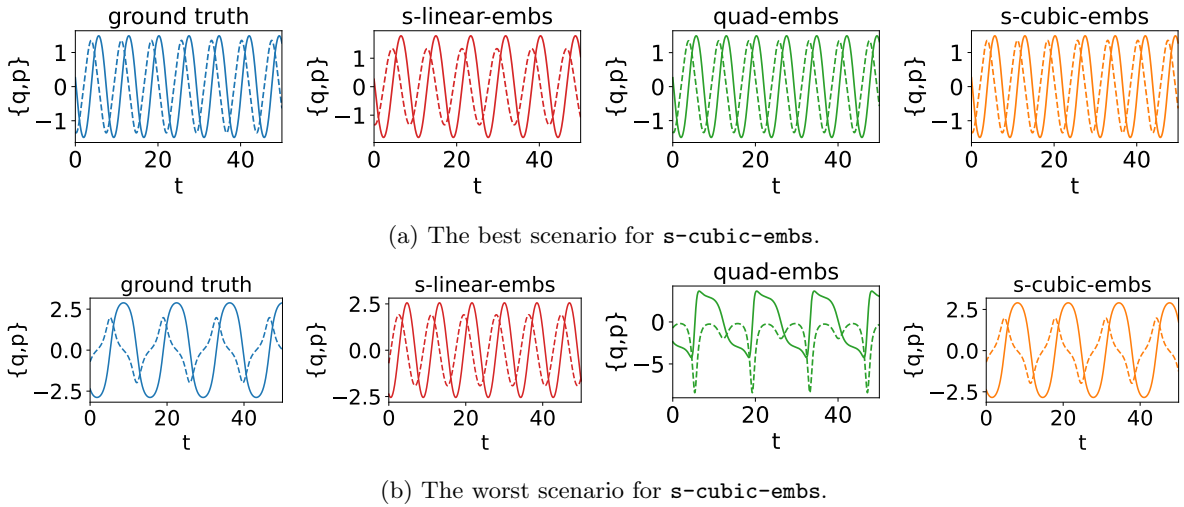


Figure 4.3: Nonlinear pendulum: Comparisons of the time-domain simulations using learned different representations test conditions for which **s-cubic-embs** performs the best and worst among the considered 25 test conditions.

dynamics, where it fails completely despite preserving the Hamiltonian for the embeddings. On the other hand, the proposed methodology, namely **s-cubic-embs**, accurately produces the dynamics for all test initial conditions. Notably, all these dynamics are stable by construction, making **s-cubic-embs** a reliable and robust approach even for nonlinear systems with continuous spectra.

For further qualitative analysis, we employ the following error measure:

$$\mathcal{E}(\mathbf{x}_0) = \frac{1}{\mathcal{N} \cdot m} \sum_{i=1}^{\mathcal{N}} \|\mathbf{x}^{\text{gt}}(t_i; \mathbf{x}_0) - \mathbf{x}^{\text{pred}}(t_i; \mathbf{x}_0)\|_2^2, \quad (4.3)$$

where  $\mathbf{x}^{\text{gt}}(t_i; \mathbf{x}_0) \in \mathbb{R}^m$  and  $\mathbf{x}^{\text{pred}}(t_i; \mathbf{x}_0) \in \mathbb{R}^m$  are, respectively, the ground truth and predicted solutions at time  $t_i$  for a given initial condition  $\mathbf{x}_0$ . Using the measure (4.3), we calculate  $\mathcal{E}$  for 25 random test initial conditions and present a violin plot in Figure 4.2. The plot clearly indicates the superior performance of **s-cubic-embs** compared to the other two considered methods. In Figure 4.3, we additionally plot time-domain simulations for two specific cases from the 25 test conditions for which **s-cubic-embs** performs the best and worst with respect to the ground truth. These figures show that for the best case, **quad-embs** also does well to capture the dynamics, but **s-linear-embs** fails to do so. Conversely, for the worst case, only **s-cubic-embs** could accurately capture the dynamics, although a slight shift in behavior is observed.

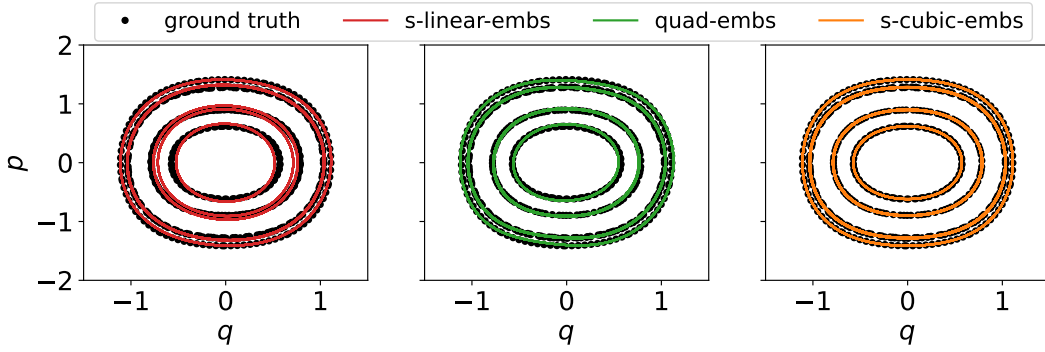


Figure 4.4: Nonlinear oscillator: The figure shows a comparison of various learned models with the ground truth in the phase space for test initial conditions.

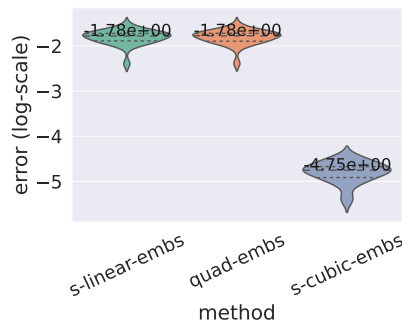


Figure 4.5: Nonlinear oscillator: A qualitative performance analysis of different methods for test initial conditions. The middle dashed line shows the median, with the precise number, across all test initial conditions.

### 4.2. Nonlinear oscillator

The second example involves a harmonic nonlinear oscillator, described by the Hamiltonian as follows:

$$\mathcal{H}(q, p) = \frac{p^2}{2} + \frac{q^2}{2} + \frac{q^2}{4}, \tag{4.4}$$

where  $q$  and  $p$  denote the position and momentum of the oscillator, respectively. To gather training data, we select 50 data points in the time interval  $[0, 4]$  for a given initial condition and consider 20 random initial conditions. We consider  $\{q, p\} \in [-2, 2] \times [-2, 2]$ . Similar to the previous example, we constrain the initial conditions so that  $\mathcal{H}(q, p) \leq 1$ .

Next, we learn suitable embeddings using **s-linear-embs**, **quad-embs**, and **s-cubic-embs** by setting the dimension of it to two. Once we have the desired embeddings, we evaluate their performance using 25 test initial conditions. Moreover, for testing, we consider a significantly longer time than the training time, which is  $[0, 50]$ , and take 5000 points in the testing time interval. First, in [Figure 4.4](#), we present the phase space plots using the learned  $\{q, p\}$  using different methods. We observe that **s-linear-embs** could not accurately capture the phase space, while **quad-embs** and **s-cubic-embs** capture it well, at least in the eyeball norm.

To conduct a qualitative comparison, we utilize the measure given in [\(4.3\)](#). Based on this, we calculate the error for each test initial condition and show them in a violin plot in [Figure 4.5](#). It shows the superior performance of **s-cubic-embs** as compared to the other two. It is worth noting that **quad-embs** and **s-cubic-embs** yield solutions that appear similar in phase space; however, **quad-embs** exhibits a significant shift in the solutions compared to **s-cubic-embs**, leading to a higher error in the measure [\(4.3\)](#) for **quad-embs**.

Furthermore, in [Figure 4.6](#), we present the time-domain simulations of two test initial conditions, where **s-cubic-embs** performs the worst and best among the considered test conditions as compared to the ground truth. We observe that **s-linear-embs** and **quad-embs** yield solutions that appear similar to the

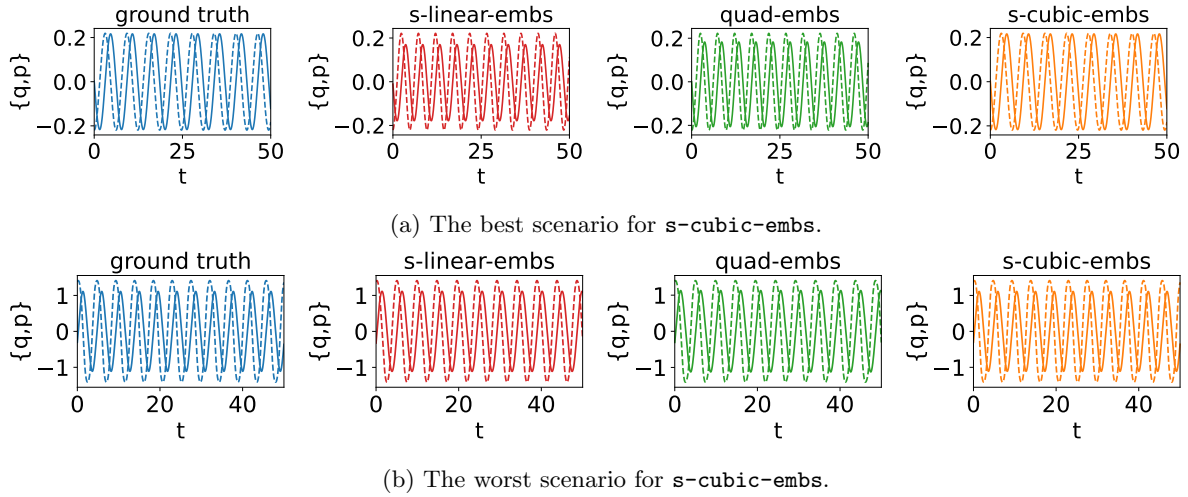


Figure 4.6: Nonlinear oscillator: Comparisons of the time-domain simulations using learned different representations test conditions for which `s-cubic-embs` performs the best and worst among the considered 25 test conditions with respect to the ground truth.

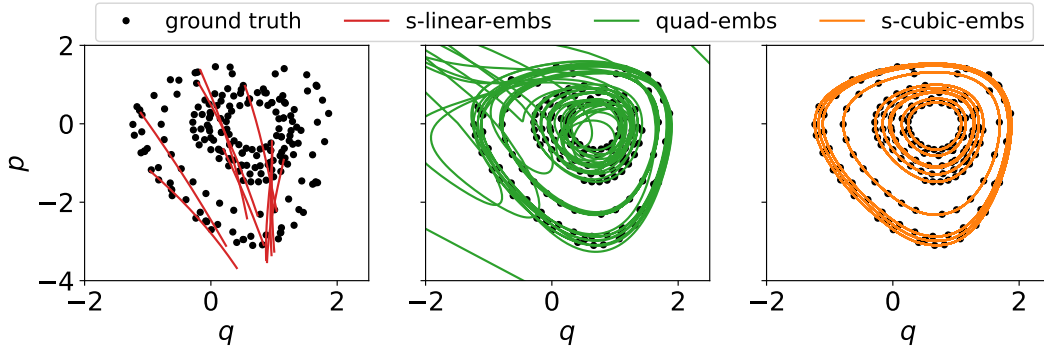


Figure 4.7: Lotka–Volterra model: The figure shows a comparison of various learned models with the ground truth in the phase space for test initial conditions.

ground truth but have a noticeable shift in the solution as compared to the ground truth. Conversely, `s-cubic-embs` outperforms both `s-linear-embs` and `quad-embs` in terms of the amplitude and period, as can be clearly seen in the solutions at around time  $t = 50$ .

### 4.3. Lotka–Volterra model

In the last low-dimensional example, we consider the Lotka–Volterra model, which is a well-known mathematical model used to describe predator-prey populations’ dynamics. It is a Hamiltonian system with the Hamiltonian as follows:

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

To learn the dynamics, we begin by collecting training data using 20 random initial conditions in the range of  $[-1.5, 1.5] \times [-1.5, 1.5]$  for  $q$  and  $p$ . From the set of initial conditions, we choose the initial conditions for which  $\mathcal{H}(q, p) \leq 4$ . For each initial condition, we consider 100 points within the time interval of  $[0, 10]$ . With the collected data, we proceed to learn the desired embeddings in a four-dimensional space using all three methods.

To assess the quality of these methods, we conducted tests using 25 different initial conditions. For the testing phase, we consider the time interval  $[0, 50]$  and take 10000 points within the testing time interval. Note that the testing time interval is five times larger than the training time interval. Like previous examples, we first plot the phase space in Figure 4.7, revealing that `s-linear-embs` does not capture the dynamics at all. This limitation could be attributed to the continuous spectra of the Lotka–Volterra model, making it challenging to derive a universal linearized model using finite-dimensional

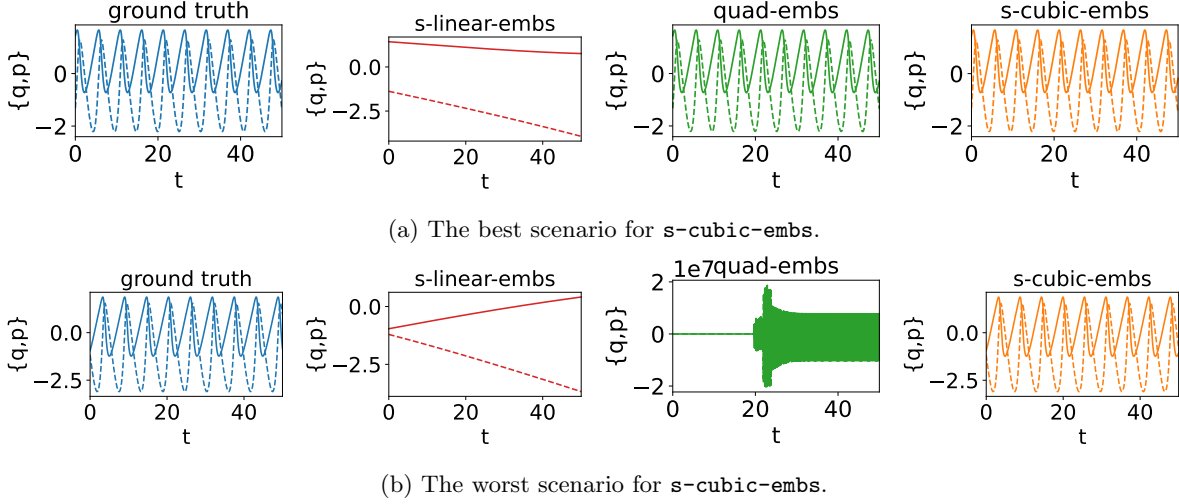


Figure 4.8: Lotka–Volterra model: Comparisons of the time-domain simulations using the learned different embeddings test conditions for which `s-cubic-embs` performs the best and worst among the considered 25 test conditions with respect to the ground truth.

embeddings. Moving forward, we evaluated `quad-embs`, which partially captured the dynamics, but it showed signs of instability when integrated over a longer time period. In contrast, `s-cubic-embs` demonstrated remarkable accuracy in learning the dynamics.

Additionally, in Figure 4.8, we plot the time-domain simulations using the test initial conditions for which `s-cubic-embs` performs the best and the worst with respect to the ground truth. The figures clearly demonstrate that `s-cubic-embs` is the only method among the three considered methods capable of accurately reproducing the dynamics for all test cases.

## 5. An Extension to High-Dimensional Data

The dynamics of various physical processes are governed by partial differential equations. As a result, the data collected for such processes are often high-dimensional. However, these data often exhibit significant spatial and temporal correlations, allowing them to represent using a low-dimensional manifold effectively. To obtain such a low-dimensional representation, we utilize the dominant POD basis obtained through an SVD of the high-dimensional training data. Furthermore, since we deal with symplectic data and assume that the underlying model is also symplectic, we incorporate the concept of co-tangent lifting [49] to determine the dominant basis using the high-dimensional snapshots, as it preserves the symplectic structure. Both the position and momenta data are concatenated in the snapshot matrix. For this, let us denote the dominant basis by  $\mathbf{V}$ . It is then followed by determining the projection matrix as  $\mathcal{V} = \begin{bmatrix} \mathbf{V} & 0 \\ 0 & \mathbf{V} \end{bmatrix}$ . Using the projection matrix, we derive a low-dimensional representation, which we refer to it as POD coordinates. The time-derivative information for these coordinates is estimated using a five-points stencil method. These coordinates are then employed for learning suitable embeddings using `s-linear-embs`, `quad-embs`, and `s-cubic-embs` to describe the dynamics of the POD coordinates. In the following, we present two examples with high-dimensional data and discuss the efficiency of the proposed methodologies.

### 5.1. Nonlinear Schrödinger equation

Towards learning dynamics using high-dimensional data, we begin by considering the nonlinear Schrödinger (NLS) equation, which has found extensive applications in analyzing the waves on the water surface and the propagation of the light in fibers. We consider the cubic Schrödinger equation given by [50]:

$$\begin{aligned}
 i \frac{\partial u}{\partial t} + \frac{1}{2} u_{\zeta\zeta} + |u|^2 u &= 0, \\
 u(0, \zeta) &= u_0(\zeta), \quad \zeta \in \Omega := [-10, 10],
 \end{aligned}
 \tag{5.1}$$

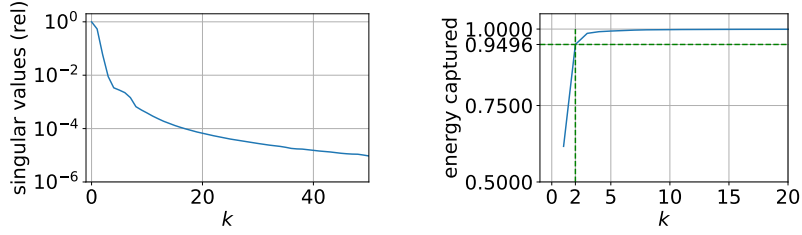


Figure 5.1: NLS example: The figure shows the decay of the singular values, obtained using the training data. The right plot indicates the energy captured using 2 dominant modes.

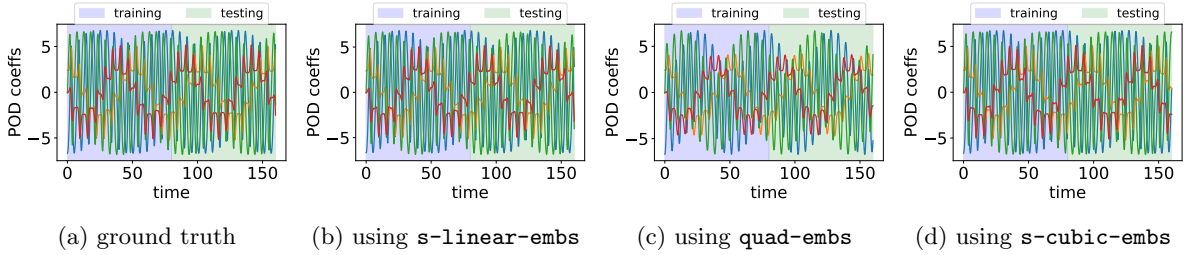


Figure 5.2: NLS model: A comparison of the transient responses of the learned models using different methods with the ground truth model.

subject to periodic boundary conditions, and  $u_{\zeta\zeta}$  denotes the double-derivative of  $u$  with respect to the space  $\zeta$ . To obtain the canonical Hamiltonian form of the NLS equation (5.1), we express the complex-valued solution  $u$  in terms of its imaginary and real parts, i.e.,  $u = q + ip$ . The Hamiltonian of the NLS equation is then given by

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

Following [43], we use a three-point central difference approximation for the  $\partial_{\zeta\zeta}$  and consider 256 points in the designated spatial domain. We collect the data using the initial condition  $u_0(\zeta) = \text{sech}(\zeta/2)$  over a time-interval  $[0, 160]$ , with a total of 3200 points within the time-interval. For our analysis, we divide the data into two halves: the first half is utilized for training purposes, while the second half is reserved for testing the performance of the different methods.

Since the data are high-dimensional ( $N = 2 \cdot 256$ ), our first step is to identify the POD coordinates. For this, we employ the SVD of the training data and observe its decay in Figure 5.1. We notice a reasonable decay, thus indicating a good low-dimensional representation using a linear projection. By considering the first two POD bases, we capture more than 94% of the energy present in the training data. As a result, we have four-dimensional POD coordinates by utilizing the co-tangent projection of the high-dimensional data.

Having obtained the POD coordinates, our next objective is to learn appropriate embeddings of dimension four using three different methodologies. In Figure 5.2, we depict the reconstruction of the POD coordinates through these identified embeddings using different methods. We observe that `quad-embs` fails to capture the dynamics in both training and testing phases, whereas `s-linear-embs` and `s-cubic-embs` perform well in capturing the high-level dynamic features. This is more apparent when the learned solutions are plotted in the phase-space in the testing phase; see Figure 5.3 (first four columns).

For a qualitative comparison, in Table 5.1, we also report the relative  $L_2$ -norm of the error for the POD coefficients for the training and testing phases. We notice that `s-linear-embs` yields the best results among the considered cases. Intuitively, one might expect `s-cubic-embs` to be as good as `s-linear-embs` since setting blocks of  $\mathbf{Q}$  in (3.12) to zero boils down to `s-linear-embs` case. However, when the dynamics are obtained using a numerical integration method, an over-fitting is observed for `s-cubic-embs`, thus leading to poorer performance than `s-linear-embs`. But the performance of `s-cubic-embs` can be improved by fine-tuning hyper-parameter, e.g., with respect to the  $L_2$ -regularization of the parameters for `s-cubic-embs`. Finding good hyper-parameters is our potential avenue for our future research.

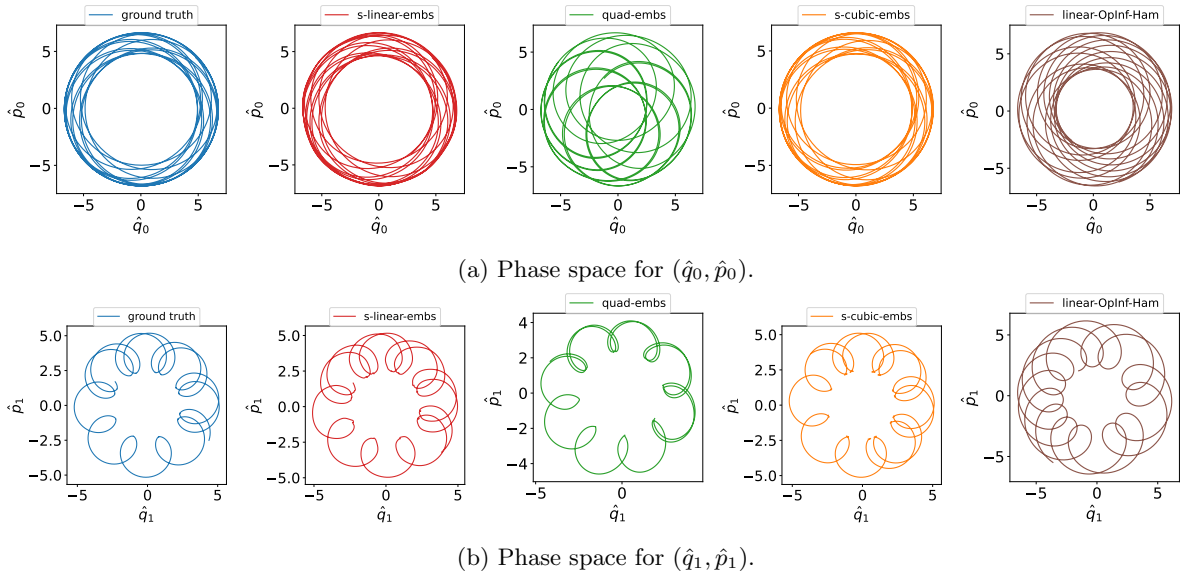


Figure 5.3: NLS model: A comparison of different methods in the testing phase. We show the positions and momenta, which are part of the POD coordinates in phase space.

	Method		
	s-linear-embs	quad-embs	s-cubic-embs
Training phase	$4.37 \cdot 10^{-3}$	$3.37 \cdot 10^{-1}$	$2.77 \cdot 10^{-1}$
Testing phase	$2.69 \cdot 10^{-2}$	$1.60 \cdot 10^0$	$1.34 \cdot 10^0$

Table 5.1: NLS model: The table shows the relative  $L_2$  errors of the POD coordinates in the training and testing phases obtained using various methods.

Furthermore, we conducted a performance comparison between `s-linear-embs` and the operator inference (`OpInf-Ham`) approach [19, 20] that also preserves the canonical Hamiltonian structure. Using `OpInf-Ham`, we also learn a linear system, aiming to describe the dynamics of the POD coordinates. In Figure 5.3, we show the results for `OpInf-Ham` (the fifth column) in the testing, clearly demonstrating that `OpInf-Ham` fails to capture the dynamics. These findings underscore the significance of learning the correct embeddings for dynamic system modeling, which we achieve, for example, for `s-linear-embs` by unleashing the power of deep learning.

Lastly, we aim at reconstructing the solution over the considered entire spatial domain using the POD coordinates. We explore three different approaches for this purpose. The first and simplest approach involves using the POD basis to reconstruct the solution, where the solution is reconstructed through a linear combination of the POD coordinates. We refer to it as *linear-decoder*. Next, we consider the quadratic-manifold approach [51], which constructs the solution on the entire spatial domain using a quadratic function of the POD coordinates. As a result, we refer to it as *quad-decoder*. Finally, we explore a convolution neural network (CNN) based approach to reconstruct the solution using the POD coordinates. We refer to it as *convo-decoder*. The architecture of the CNN is discussed in Appendix C, and therein, the training of both the quadratic and CNN-based decoders is also provided.

Using these decoder approaches, we plot the reconstructions of the solution of the full spatial domain in Figure 5.4. Note that we have used the learned embeddings, obtained using `s-linear-embs` as it performs the best among the considered ones (see Table 5.1). Upon evaluation, we notice that all three decoder approaches demonstrate reasonable performance. Particularly, we emphasize that the linear-decoder approach also performs well since more than 94% of the energy has been captured by the selected POD modes. Moreover, we notice that the quadratic decoder works slightly worse than the linear decoder in terms of solution reconstruction in the testing phase. It could be due to the over-fitting of the quad-decoder despite utilizing the  $L_2$  regularizer. On the other hand, the convo-decoder emerges as the most effective method among those considered (notice the limits of the bar-plots in Figure 5.4 and Table 5.2).



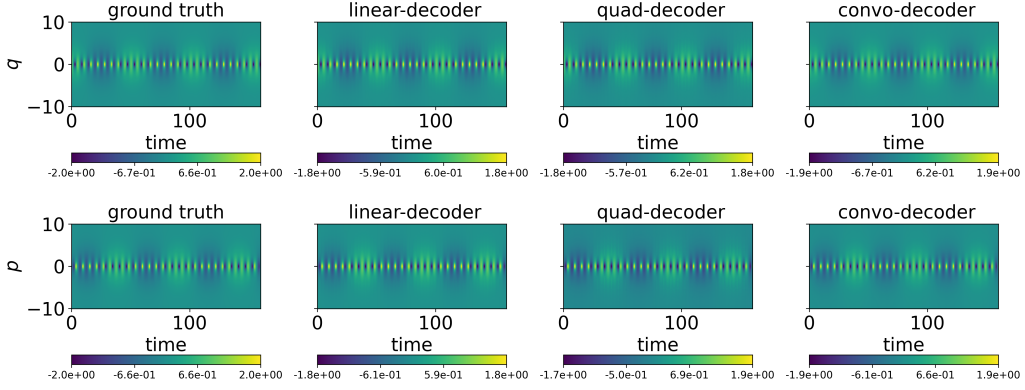


Figure 5.4: NLS example: A comparison of the reconstructions of  $\mathbf{q}$  and  $\mathbf{p}$  using the learned POD coordinates  $\hat{\mathbf{q}}$  and  $\hat{\mathbf{p}}$  using `s-linear-embs` by means of various decoder methods.

	Methods		
	linear-decoder	quad-decoder	convo-decoder
Training phase	$7.10 \cdot 10^{-4}$	$7.40 \cdot 10^{-4}$	<b><math>4.62 \cdot 10^{-4}</math></b>
Testing phase	$3.00 \cdot 10^{-3}$	$3.12 \cdot 10^{-3}$	<b><math>2.78 \cdot 10^{-3}</math></b>

Table 5.2: NLS model: The table shows the mean  $L_2$ -errors of the reconstructed and ground truth solutions using various decoder approaches.

## 5.2. Wave example

In our last example, we consider a linear wave equation of the form:

$$\begin{aligned} u_{tt} &= u_{\zeta\zeta}, \\ u(t_0, \zeta) &= u_0(\zeta), \quad \zeta \in \Omega := [-10, 10], \end{aligned} \tag{5.2}$$

with a periodic boundary condition. The wave equation is an example of a Hamiltonian PDE [52], and its corresponding Hamiltonian is defined as

$$\mathcal{H}(u) = \frac{1}{2} \int_{\Omega} (q_{\zeta}^2 + p^2) d\zeta,$$

where  $p = u_t$  and  $q = u$ , and  $q_{\zeta}$  denotes the partial derivative of  $q$  with respect to  $\zeta$ . Discretizing the PDE as discussed in [43], we can arrive at a discretized model as follows:

$$\frac{d\mathbf{z}}{dt} = \mathbf{K}\mathbf{z}, \quad \text{with } \mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} \quad \text{and } \mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_N \\ \mathbf{D}_{\zeta\zeta} & \mathbf{0} \end{bmatrix}. \tag{5.3}$$

In the above equation,  $\mathbf{D}_{\zeta\zeta} \in \mathbb{R}^{N \times N}$  denotes an approximation of  $\partial_{\zeta\zeta}$ ,  $\mathbf{0} \in \mathbb{R}^{N \times N}$  and  $\mathbf{I}_N \in \mathbb{R}^{N \times N}$  are the zero and identity matrices, and  $(\mathbf{q}, \mathbf{p})$  are the discretized  $(q, p)$ , and  $N$  denotes the number of grid points in the space.

Next, we generate training data using a parameterized initial condition  $u_0(\zeta; \mu)$ , where  $u_0(\zeta; \mu) = \text{sech}(\mu \cdot \zeta)$  with  $\mu$  ranging from 0.5 to 1.4. We collect data using 10 different values of  $\mu$ , i.e.,  $\mu = \{0.5, 0.6, \dots, 1.3, 1.4\}$ . Among these, we designate three values of  $\mu$  ( $\mu = \{0.7, 1.0, 1.2\}$ ) for testing, while the remaining seven values serve as training data. Moreover, we consider 256 equidistant points in the spatial domain  $\Omega = [-10, 10]$ , resulting in a model of dimension 512. For each training initial condition, we consider 501 points in the time interval  $[0, 25]$ .

As considered in the previous example, we first aim to project the high-dimensional data onto a low-dimensional subspace and obtain the POD coordinates. With that goal, in Figure 5.5, we examine the decay of the singular values, indicating a relatively slow decay of it. It is expected as the considered system has a transport phenomenon, which often has a slow-decay of Kolmogorov  $N$ -width [53, 54]. Next, we consider the three dominant modes to determine the projection matrix, which captures only 52.7% of the energy present in the training data. We obtain the POD coordinates using the principle of

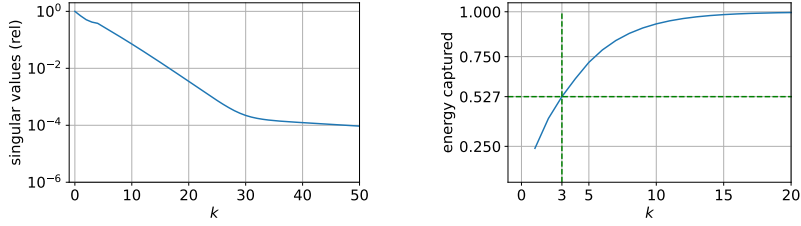


Figure 5.5: Wave example: The figure shows the delay of the singular values obtained using the training data. The right plot shows the energy captured using 3 dominant modes.

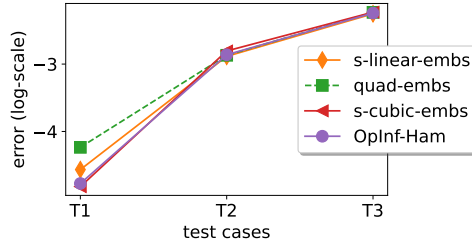


Figure 5.6: Wave example: A comparison of all three considered methodologies on test cases.

co-tangent lifting for canonical Hamiltonian systems, encompassing the reduced momenta and position, respectively, denoted by  $\hat{\mathbf{p}} \in \mathbb{R}^3$  and  $\hat{\mathbf{q}} \in \mathbb{R}^3$ .

Next, using  $\hat{\mathbf{q}}$  and  $\hat{\mathbf{p}}$ , we learn the desired embeddings using `s-linear-embs`, `quad-embs`, and `s-cubic-embs` to learn the dynamics of the POD coordinates through these embeddings. For comparison, we also include the `OpInf-Ham` approach, which we discussed in the previous example. To evaluate the performance of these approaches, we utilize three left-out test initial conditions. Note that the test initial conditions and the dynamics evolution using the test initial conditions are high-dimensional, which are then projected using the same projection matrix determined in the training phase to obtain the POD coordinates. In Figure 5.6, we present a qualitative evaluation of their performances based on the error measure (4.3). It is interesting to note that all methods show remarkably similar performance. Surprisingly, `OpInf-Ham` with linear models is able to capture the dynamics of the POD coordinates well, indicating that there is no particular need to learn a nonlinear coordinate transformation since the POD coordinates themselves are good, and their dynamic evaluation can be accurately learned using a linear system.

Next, using the POD coordinates, we focus on constructing the solution on the full-spatial domain using the POD coordinates. Similar to the previous example, we employ three methods: linear-decoder, quad-decoder, and convo-decoder. Although all four approaches, namely `s-linear-embs`, `quad-embs`, `s-cubic-embs`, and `OpInf-Ham` perform competitively to learn the dynamics for POD coordinates, we use the recovered POD coordinates using `s-linear-embs`, to be consistent with the previous example. For two of the test scenarios, in Figure 5.7, we present the reconstructed solution on the entire domain using various decoder approaches. In Table 5.3, we also report the mean  $L_2$ -errors between different reconstructed solutions and the ground truth.

Our observation reveals that the linear-decoder performs very poorly, which is to be expected, considering that the data has slow-decaying Kolmogorov  $N$ -widths [54] and the three POD basis capture only 52.7% of the energy present in the training data. With the quad-decoder, the quality of the reconstruction improves as compared to the linear-decoder but still exhibits artifacts; see the third column in Figure 5.7. On the other hand, the convo-decoder outperforms the other two considered approaches to reconstruct the solution, uncovering the power of neural networks in this context and helping in breaking Kolmogorov  $N$ -width barrier [55].

## 6. Conclusions

To conclude, we have explored the capabilities of deep learning to learn embeddings that aim to provide a universal linearized framework for nonlinear canonical Hamiltonian systems. It can be viewed as a

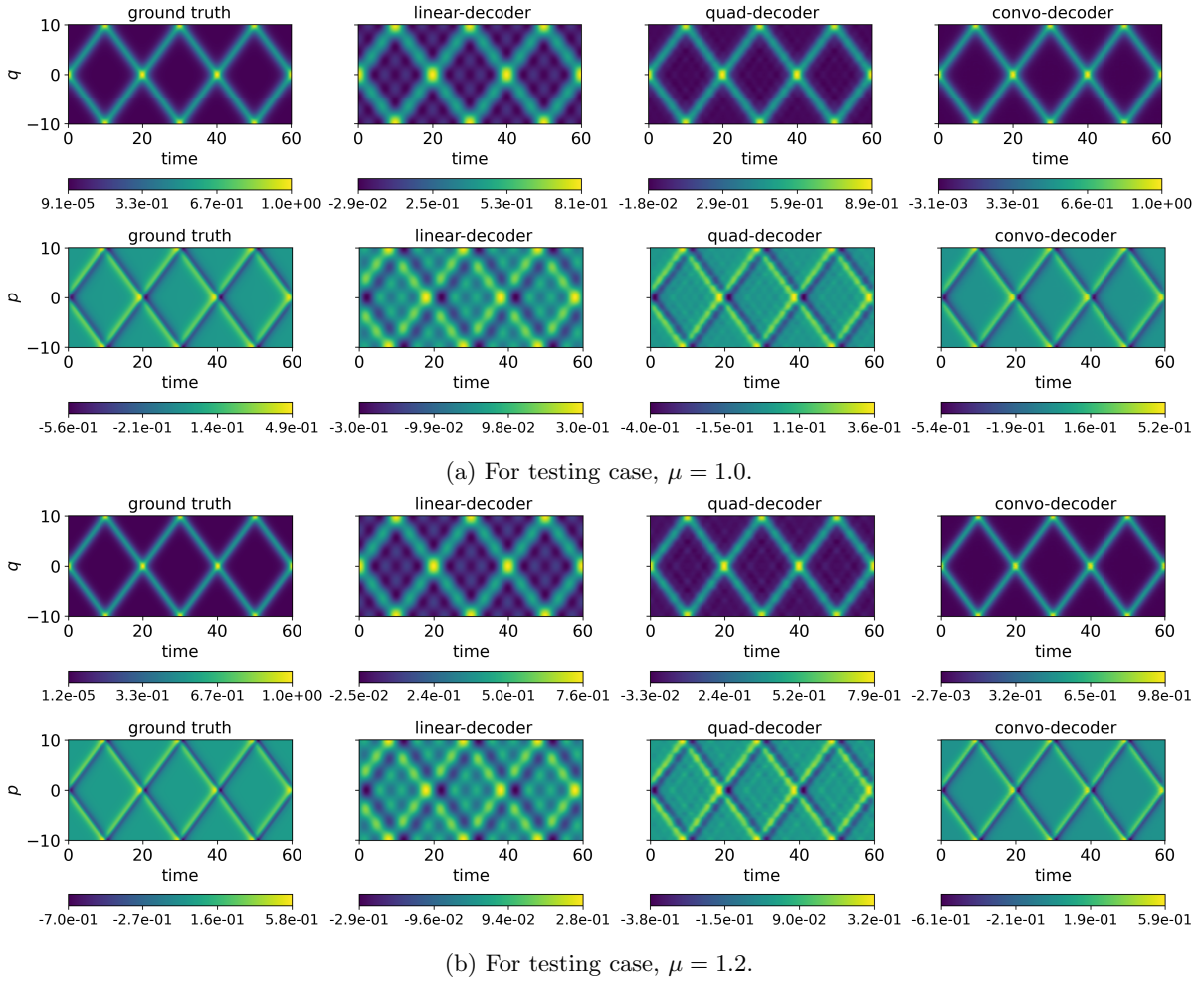


Figure 5.7: Wave example: A comparison of the reconstruction of  $\mathbf{q}$  and  $\mathbf{p}$  using the POD coordinates  $\hat{\mathbf{q}}$  and  $\hat{\mathbf{p}}$ .

structure-preserving finite-dimensional Koopman operator learning for nonlinear canonical Hamiltonian systems. Designing embeddings with the desired properties is inherently challenging. To address this, we employed the autoencoder framework, which proved to be instrumental in achieving our objectives. In addition, we have overcome the challenges associated with learning global linearization for nonlinear systems featuring a continuous eigenvalue spectrum by learning a global cubicized framework. It is inspired by the lifting principle [44, 45], allowing us to express nonlinear systems as cubic systems in a finite-dimensional embedding. Furthermore, we delved into how to enforce stability with regard to the dynamics of the learned embeddings, and it is achieved by exploring a sum-of-squares formulation for the underlying Hamiltonian. We have demonstrated the efficiency of the proposed methodologies for low-dimensional cases. We additionally have discussed a way to deal with high-dimensional data. To handle such cases, we utilized proper-orthogonal decomposition (POD) to derive a low-dimensional representation of these high-dimensional data. We then leveraged this representation to learn a suitable embedding with the desired properties. Moreover, from these low-dimensional representations obtained using POD, we investigated various decoder approaches, namely, linear-decoder, quadratic-decoder, and convolutional neural network-based decoder, to reconstruct the solutions on the full spatial domain. Notably, we observed that convolutional neural networks can perform well when the POD basis does not capture enough of the energy present in the training data. Hence, with a nonlinear decoder based on a convolutional neural network, a good reconstruction with only a few POD modes can be done for systems with a slow decay of the Kolmogorov  $N$ -width. Lastly, we emphasize that incorporating prior knowledge about the canonical Hamiltonian framework played a crucial role in enabling us to learn dynamics in low-data regimes. It also enhances the interpretability of the learned coordinate system in the sense that it obeys the properties of the canonical Hamiltonian system by design.

This work opens up several challenging and important research avenues for the future. The foremost

	Methods		
	linear-decoder	quad-decoder	convo-decoder
Test for $\mu = 0.7$	$2.01 \cdot 10^{-3}$	$1.88 \cdot 10^{-4}$	<b><math>1.40 \cdot 10^{-5}</math></b>
Test for $\mu = 1.0$	$5.26 \cdot 10^{-3}$	$9.54 \cdot 10^{-4}$	<b><math>1.06 \cdot 10^{-4}</math></b>
Test for $\mu = 1.2$	$9.18 \cdot 10^{-3}$	$2.84 \cdot 10^{-3}$	<b><math>4.99 \cdot 10^{-4}</math></b>

Table 5.3: Wave model: The table shows the mean  $L_2$ -errors of the reconstructed and ground truth solutions using various decoder approaches.

challenge is to design an autoencoder architecture and to determine the dimension of the coordinate system  $\mathbf{y}$ . Automating the design of the autoencoder architecture using neural architecture search, see, e.g., [56], would be highly beneficial, as well as exploring methods to determine the optimal dimensionality of the coordinate system. Additionally, in practical scenarios, the presence of noise in data is inevitable. Therefore, it will be crucial to adapt the proposed methodologies to handle such cases, and for this, one can employ the principles discussed, e.g., in [57, 58]. Furthermore, for high-dimensional data, we have determined a low-dimensional representation of these data by using POD, which can be seen as a linear projection or transformation. However, an investigation based on convolution neural networks to determine a low-dimensional projection would be worthwhile. In the future, we also aim to apply these techniques to practical problems, particularly those arising in the study of planetary motion.

## References

- [1] B. Leimkuhler and S. Reich, “Simulating Hamiltonian mechanics,” *Cambridge Monogr. Appl. Comput. Math.*, 2005.
- [2] J. E. Marsden and T. S. Ratiu, *Introduction to mechanics and symmetry: A basic exposition of classical mechanical systems*, vol. 17. Springer Science & Business Media, 2013.
- [3] V. I. Arnol’d, *Mathematical Methods of Classical Mechanics*, vol. 60. Springer Science & Business Media, 2013.
- [4] S. N. Kumpati and P. Kannan, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, 1990.
- [5] R. Rico-Martinez, J. Anderson, and I. Kevrekidis, “Continuous-time nonlinear signal processing: a neural network based approach for gray box identification,” in *Proc. IEEE Workshop on Neural Netw. for Signal Processing*, pp. 596–605, IEEE, 1994.
- [6] P. Van Overschee and B. de Moor, *Subspace Identification of Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.
- [7] J. A. Suykens, J. P. Vandewalle, and B. L. de Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Springer, 1996.
- [8] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, NJ, 1999.
- [9] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proc. Nat. Acad. Sci. U.S.A.*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [10] B. Peherstorfer and K. Willcox, “Data-driven operator inference for noninvasive projection-based model reduction,” *Comp. Meth. Appl. Mech. Eng.*, vol. 306, pp. 196–215, 2016.
- [11] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Adv. Neural Inform. Process. Sys.*, pp. 6571–6583, 2018.
- [12] 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.*, vol. 378, pp. 686–707, 2019.

- [13] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” *Advances in neural information processing systems*, vol. 32, 2019.
- [14] Y. D. Zhong, B. Dey, and A. Chakraborty, “Dissipative SYMODEN: Encoding Hamiltonian dynamics with dissipation and control into deep learning,” *arXiv preprint arXiv:2002.08860*, 2020.
- [15] Y. D. Zhong, B. Dey, and A. Chakraborty, “Symplectic ODE-NET: Learning Hamiltonian dynamics with control,” *arXiv preprint arXiv:1909.12077*, 2019.
- [16] S. A. Desai, M. Mattheakis, D. Sondak, P. Protopapas, and S. J. Roberts, “Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems,” *Phys. Rev. E*, vol. 104, no. 3, p. 034312, 2021.
- [17] T. Duong and N. Atanasov, “Hamiltonian-based neural ODE networks on the  $SE(3)$  manifold for dynamics learning and control,” *arXiv preprint arXiv:2106.12782*, 2021.
- [18] P. Jin, Z. Zhang, I. G. Kevrekidis, and G. E. Karniadakis, “Learning Poisson systems and trajectories of autonomous systems via Poisson neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.
- [19] H. Sharma, Z. Wang, and B. Kramer, “Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems,” *Physica D*, vol. 431, p. 133122, 2022.
- [20] A. Gruber and I. Tezaur, “Canonical and noncanonical Hamiltonian operator inference,” *arXiv preprint arXiv:2304.06262*, 2023.
- [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,” 2023. arXiv:2305.15490.
- [22] H. Sharma and B. Kramer, “Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale mechanical systems,” 2022. arXiv:2203.06361.
- [23] B. O. Koopman, “Hamiltonian systems and transformation in Hilbert space,” *Proc. Nat. Acad. Sci. U.S.A.*, vol. 17, no. 5, pp. 315–318, 1931.
- [24] B. O. Koopman and J. V. Neumann, “Dynamical systems of continuous spectra,” *Proc. Natl. Acad. Sci. USA*, vol. 18, no. 3, pp. 255–263, 1932.
- [25] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson, “Spectral analysis of nonlinear flows,” *J. Fluid Mech.*, vol. 641, no. 1, pp. 115–127, 2009.
- [26] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” *J. Fluid Mech.*, vol. 656, pp. 5–28, 2010.
- [27] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. Philadelphia, PA: SIAM Publications, 2016.
- [28] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition,” *J. Nonlinear Sci.*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [29] F. Noé and F. Nüske, “A variational approach to modeling slow processes in stochastic dynamical systems,” *Multiscale Modeling & Simulation*, vol. 11, no. 2, pp. 635–655, 2013.
- [30] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. Mey, and F. Noé, “Variational approach to molecular kinetics,” *J. Chem. Theory Comput.*, vol. 10, no. 4, pp. 1739–1752, 2014.
- [31] F. Nüske, R. Schneider, F. Vitalini, and F. Noé, “Variational tensor approach for approximating the rare-event kinetics of macromolecular systems,” *J. Chem. Phys.*, vol. 144, no. 5, 2016.
- [32] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, “Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control,” *PloS One*, vol. 11, no. 2, p. e0150171, 2016.
- [33] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, “A kernel-based method for data-driven Koopman spectral analysis,” 2015.

- [34] C. Wehmeyer and F. Noé, “Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics,” *J. Chem. Phys.*, vol. 148, no. 24, 2018.
- [35] A. Mardt, L. Pasquali, H. Wu, and F. Noé, “VAMPnets for deep learning of molecular kinetics,” *Nat. Commun.*, vol. 9, no. 1, p. 5, 2018.
- [36] N. Takeishi, Y. Kawahara, and T. Yairi, “Learning Koopman invariant subspaces for dynamic mode decomposition,” in *Adv. Neural Inform. Process. Sys.* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, pp. 1130–1140, 2017.
- [37] E. Yeung, S. Kundu, and N. Hodas, “Learning deep neural network representations for Koopman operators of nonlinear dynamical systems,” in *American Control Conference (ACC)*, pp. 4832–4839, IEEE, 2019.
- [38] S. E. Otto and C. W. Rowley, “Linearly recurrent autoencoder networks for learning dynamics,” *SIAM J. Appl. Dyn. Syst.*, vol. 18, no. 1, pp. 558–593, 2019.
- [39] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, “Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator,” *Chaos*, vol. 27, no. 10, 2017.
- [40] 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.
- [41] J. Zhang, Q. Zhu, and W. Lin, “Hamiltonian neural Koopman operator,” in *Workshop paper at the Symbiosis of Deep Learning and Differential Equations II—NeurIPS*, 2022. Available at <https://openreview.net/forum?id=oeIr0pv-sMw>.
- [42] P. Buchfink, S. Glas, and B. Haasdonk, “Symplectic model reduction of Hamiltonian systems on nonlinear manifolds and approximation with weakly symplectic autoencoder,” *SIAM J. Sci. Comput.*, vol. 45, no. 2, pp. A289–A311, 2023.
- [43] S. Yildiz, P. Goyal, T. Bendokat, and P. Benner, “Data-driven identification of quadratic symplectic representations of nonlinear Hamiltonian systems,” *arXiv preprint arXiv:2308.01084*, 2023.
- [44] M. A. Savageau and E. O. Voit, “Recasting nonlinear differential equations as S-systems: a canonical nonlinear form,” *Mathematical Biosciences*, vol. 87, no. 1, pp. 83–115, 1987.
- [45] 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.
- [46] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, “Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems,” *Phys. D*, vol. 406, p. 132401, 2020.
- [47] P. Goyal and P. Benner, “Generalized quadratic-embeddings for nonlinear dynamics using deep learning,” *arXiv preprint arXiv:2211.00357*, 2022.
- [48] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, vol. 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2002.
- [49] L. Peng and K. Mohseni, “Symplectic model reduction of Hamiltonian systems,” *SIAM J. Sci. Comput.*, vol. 38, no. 1, pp. A1–A27, 2016.
- [50] B. Karasözen and M. Uzunca, “Energy preserving model order reduction of the nonlinear Schrödinger equation,” *Advances in Computational Mathematics*, vol. 44, no. 6, pp. 1769–1796, 2018.
- [51] R. Geelen, S. Wright, and K. Willcox, “Operator inference for non-intrusive model reduction with quadratic manifolds,” *Comp. Meth. Appl. Mech. Eng.*, vol. 403, p. 115717, 2023.
- [52] T. J. Bridges and S. Reich, “Numerical methods for Hamiltonian PDEs,” *Journal of Physics A: mathematical and general*, vol. 39, no. 19, p. 5287, 2006.
- [53] A. Kolmogoroff, “Über die beste annäherung von funktionen einer gegebenen Funktionenklasse,” *Annals of Mathematics*, pp. 107–110, 1936.



- [54] C. Greif and K. Urban, “Decay of the Kolmogorov N-width for wave problems,” *Appl. Math. Lett.*, vol. 96, pp. 216–222, 2019.
- [55] B. Peherstorfer, “Breaking the Kolmogorov barrier with nonlinear model reduction,” *Notices Amer. Math. Soc.*, vol. 69, no. 5, pp. 725–733, 2022.
- [56] T. Elsken, J. H. Metzen, and F. Hutter, “Neural architecture search: A survey,” *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.
- [57] S. H. Rudy, J. N. Kutz, and S. L. Brunton, “Deep learning of dynamics and signal-noise decomposition with time-stepping constraints,” *J. Comput. Phys.*, vol. 396, pp. 483–506, 2019.
- [58] 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.
- [59] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [60] S. Jain, P. Tiso, J. B. Rutzmoser, and D. J. Rixen, “A quadratic manifold for model order reduction of nonlinear structural dynamics,” *Computers & Structures*, vol. 188, pp. 80–94, 2017.

## A. Demonstrating Examples

**Example 1** (Cubic Hamiltonian). Consider a nonlinear system with the Hamiltonian  $\mathcal{H}(q, p) = \frac{p^2}{2} + \frac{q^2}{2} + \frac{q^3}{3}$ . The associated governing equations for the oscillator are thus given by

$$\begin{aligned} \dot{q} &= p \\ \dot{p} &= -q - q^2. \end{aligned} \tag{A.1}$$

It is easy to see that the system (A.1) is not globally stable. For large negative values for  $q$  and  $p$ , it can be observed that  $p \rightarrow -\infty$  and  $q \rightarrow -\infty$  as  $t \rightarrow \infty$ , despite  $\mathcal{H}(q, p)$  being constant. Its stability can also be argued by the fact that the Hamiltonian is neither bounded from below nor from above.

**Example 2.** Consider a system with a Hamiltonian  $\mathcal{H}(p, q) = p^2 + q^4$ . If we aim to write the Hamiltonian in the form given in (3.13), then we have

$$\mathcal{H}(p, q) = \begin{bmatrix} q \\ p \\ q^2 \\ qp \\ pq \\ p^2 \end{bmatrix}^\top \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} q \\ p \\ q^2 \\ qp \\ pq \\ p^2 \end{bmatrix} \tag{A.2}$$

It can be noted that  $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ . To decompose  $\mathbf{Q}$  to write in the form given in (3.14), we define

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{A.3}$$

First, note that  $\mathbf{V}^\top \mathbf{Q} \mathbf{V} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ , where  $\mathbf{Q}_1 = \mathbf{I}_2$  and  $\mathbf{0}$  is a zero-matrix of the appropriate size.

Furthermore, notice that  $\left\| \mathbf{V} \begin{bmatrix} \mathbf{z} \\ \mathbf{z} \otimes \mathbf{z} \end{bmatrix} \right\|_2 = q^4 + p^2 =: \mathbf{g}(\mathbf{z})$ , where  $\mathbf{z} = [q, p]$ . It can be noted that  $\mathbf{g}(\mathbf{z})$  is non-negative, and for  $\|\mathbf{z}\| \rightarrow \infty$ ,  $\mathbf{g}(\mathbf{z}) \rightarrow \infty$ . Since  $\mathbf{g}(\mathbf{z})$  is finite constant (as it is a Hamiltonian, corresponding to the underlying dynamical system),  $\|\mathbf{z}\|$  cannot be infinite; otherwise,  $\mathbf{g}(\mathbf{z})$  must also be infinite. Using these arguments, the dynamics will be stable, and  $\|\mathbf{z}(t)\|_2$  will be bounded  $\forall t \geq 0$ .

Examples/Parameters	Encoder layers [neurons]	Coordinate dimension	Batch size	weight decay ( $\text{wd}^{(a)}$ , $\text{wd}^{(h)}$ )
Nonlinear pendulum	[8, 8, 8]	2	32	( $10^{-5}$ , $10^{-5}$ )
Nonlinear oscillator	[8, 8, 8]	2	32	( $10^{-5}$ , $10^{-5}$ )
Lotka-Volterra	[8, 8, 8]	4	64	( $10^{-5}$ , $10^{-4}$ )
Nonlinear Schrodinger	[12, 12, 12]	4	32	( $10^{-5}$ , $10^{-3}$ )
Linear Wave	[12, 12, 12]	6	32	( $10^{-5}$ , $10^{-5}$ )

Table B.1: The table contains all the hyper-parameters to learn the dynamics of the considered examples. We note that  $\text{wd}^{(a)}$  and  $\text{wd}^{(h)}$ , respectively, denote weight decay parameters for the learnable parameters for the autoencoder and Hamiltonian for the transformed coordinate system.

## B. Training Set-up

First note that all the experiments are done using PyTorch on a machine with an Intel® Core™ i5-12600K CPU and NVIDIA RTX™ A4000(32GB) GPU. We set the number of epochs for each example to 4000 and the initial learning rate to  $3 \cdot 10^{-3}$ . We have used the Adam optimizer [59] to train the parameters for the autoencoder and for the Hamiltonian, which defines the dynamics of the learned coordinate system or embedding. Furthermore, we utilize a step decay for the learning rate, using the StepLR implementation in PyTorch with a step of 0.1. For each experiment, we set the hyper-parameters  $(\lambda_1, \lambda_2, \lambda_3)$  in (3.7) to  $(0.1, 1.0, 1.0)$ . For the autoencoder architecture, we use a multi-layer perceptron (MLP) architecture with three hidden layers and SeLU activation, and their number of neurons are given in Table B.1. The table also contains additional hyper-parameters for our illustrative examples. Note that all these parameters are derived based on a combination of experience and quick parameter grid-search, but a systematic and automatic way to determine them remains a future research topic. Furthermore, we control the parameters, defining the Hamiltonian for the coordinate system, by penalizing using the weighted  $L_1$ -norm, and the weight for all examples is set to  $10^{-4}$ .

## C. Quadratic-Decoder and Convolutional Neural Network-Based Decoder

Here, we discuss the construction of solutions on the designated spatial domain using the low-dimensional POD coordinates. A simple approach is a linear decoder in which the POD basis are used to reconstruct the solution. In the following, we discuss quadratic-decoder and convolutional neural network (CNN) based decoder.

**Quadratic-decoder:** Following [51, 60], we pose the following optimization problem, aiming to reconstruct the solution on the entire domains:

$$\min_{\mathbf{V}, \mathbf{H}} \sum_{i=1}^{\mathcal{N}} \|\mathbf{x}_i - \mathbf{V}\mathbf{y}_i - \mathbf{H}(\mathbf{y}_i \otimes \mathbf{y}_i)\|, \quad (\text{C.1})$$

where  $\mathbf{x}_i$  and  $\mathbf{y}_i$  are the  $i$ th instance of the full domain solution and the corresponding POD coordinates, respectively and the norm  $\|\cdot\|$  in the above equation is a combination of Frobenius norm and  $L_1$ -norm with an equal weight-age. Since a quadratic function is to reconstruct the full-domain solution, we refer to it as a quad-decoder.

**CNN based decoder:** Benefiting with the success of deep learning, we also propose to reconstruct the full-domain solutions by means of CNNs. It takes the POD coordinates as inputs and provides an output as the full-domain solution. Precisely, we set up the following optimization problem:

$$\min_{\eta} \sum_{i=1}^{\mathcal{N}} \|\mathbf{x}_i - \mathcal{N}_{\eta}(\mathbf{y}_i)\|, \quad (\text{C.2})$$

where  $\mathcal{N}$  is a CNN, and its trainable parameters are denoted by  $\eta$ , and the norm  $\|\cdot\|$  in the above equation is a combination of Frobenius norm and  $L_1$ -norm with an equal weight-age. We propose the

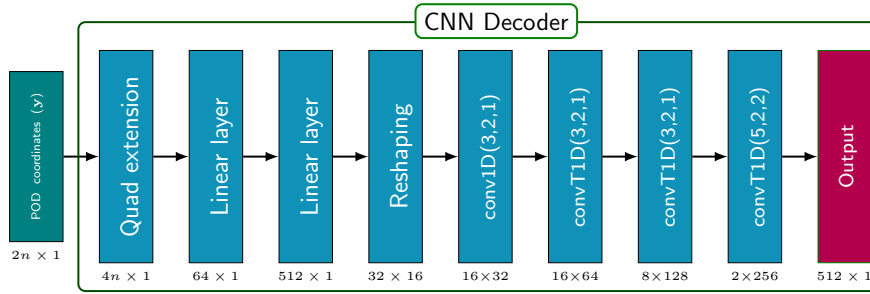


Figure C.1: The figure summarises the architecture of the CNN decoder. In the figure, the quad-extension block concatenates the input and its Hadamard product, i.e.,  $[\mathbf{y}, \mathbf{y} \circ \mathbf{y}]$ , and  $\text{convT1D}(k, s, p)$  is a 1D transpose convolution layer with transpose kernel size  $k$ , stride size  $s$ , padding size  $p$ . We have used an output padding size of 1 to obtain a symmetric auto-encoder structure in 1D transpose convolution layers. We denote the size of the output block below each block.

architecture of a CNN as in [Figure C.1](#). The parameters of these two decoders are optimized using the Adam optimizer for 600 epochs. We also set the initial learning rate to  $10^{-3}$ , which is reduced by a factor of ten after each 250 epochs. We also use weight-decay with a weightage of  $10^{-5}$  to avoid over-fitting. Moreover, we have set the batch-size as 32.