UNIVERSITY OF TARTU
Institute of Physics
Physics, chemistry and materials science

Gordei Pribõtkin

Hamiltonian Neural Networks for coupled Duffing oscillators

Bachelor's Thesis (6 ECTS)

Supervisor: Dr. Stefania Tomasiello

Tartu 2022

Hamiltonian Neural Networks for coupled Duffing oscillators

Abstract: In this thesis, the performances of two physics-guided neural network approaches, namely the Hamiltonian Neural Network (HNN) and the port-Hamiltonian Neural Network (pHNN), are analysed through the predicted dynamics of coupled Duffing oscillators and compared to a baseline non-Hamiltonain approach. We use HNN to model systems of no more than five coupled unforced and undamped Duffing oscillators, and explore how the model generalises to different initial conditions. The pHNN approach is considered for a system of two coupled Duffing oscillators with forcing and damping terms, where we also explore alternative activation functions in one of the layers. We find that HNN approach yielded generally good models for the systems we considered. The pHNN model that incorporated sinusoidal activation function resulted in good models, but failed to extrapolate in time, while the models with other activation functions failed to give accurate predictions.

Keywords:
Port-Hamiltonian, Canonical equations, Periodic orbits, Neural Networks

CERCS: P170 Computer science, numerical analysis, systems, control

Hamiltoni närvivõrgud sidestatud Duffingu ostsillaatoritele

Lühikokkuvõte: Käesolevas bakalaureusetöös analüüsitakse kahte füüsika-juhitud närvivõrgupõhiseid lähenemisviise dünaamiliste süsteemide modelleerimiseks, nimelt, Hamiltoni tehisnärvivõrke (HNN) ja port-Hamiltoni tehisnärvivõrke (pHNN). HNN lähenemisviis on rakendatud süsteemidele, mis sisaldavad kuni viit sidestatud summutamata ja sundimata Duffingu ostsillaatoreid. pHNN lähenemisviise vaadeldakse kahest sunnitud ja summutatud perioodilise käitumisega ostsillatorite baasil, kus me uurime ka alternatiivseid aktiveerimisfunktsioone ühes närvivõrku kihis. Leiame, et HNN lähenemisviis andis häid mudeleid enamiku vaadeldud süsteemide jaoks ning mudelid üldistusid uutele algtingimustele. pHNN-mudel, mis sisaldas sinusoidset aktiveerimisfunktsiooni, andis häid mudeleid, kuid ei suutnud ekstrapoleerida ajas. Teiste aktiveerimisfunktsioonidega parandanud mudelitest saadud prognoosid on olnud väga ebatäpsed kogu vaadeldud ajavahemikus.

Võtmesõnad:
Port-Hamiltoni formalism, Kanoonilised võrrandid, Perioodilised trajektoorid, Tehisnärvivõrgud

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

List of Tables

1 Base parameters for Duffing oscillators 20
2 Initial conditions for forced Duffing oscillators 21
3 Comments on the predictions obtained with HNN models for time-evolution of coordinate x_1 22
4 Mean total errors for different systems, models and initial conditions, presented with 2 SE uncertainty. 23

List of Figures

1 Structure of feedforward neural network with densely connected layers, n_i is the number of neurons in a i-th layer. 10
2 Structure of a Hamiltonian neural network 17
3 Structure of a port-Hamiltonian neural network 18
4 Structure of a Baseline neural network 18
5 Mean total errors for testing initial conditions 22
6 Mean total errors for validation initial conditions 22
7 Heat map of MSE for predicted derivatives of canonical coordinates at each for points on a grid in phase space ($N = 1$) 24
8 Heat map of MSE for predicted trajectories in timespan of $t \in [0, 10]$ with each points on a grid in as initial conditions, along with sampled trajectories for $t \in [0, 2.9]$ ($N = 1$) 24
9 Predicted and True x_1 vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-$\tanh$, HNN-$\tanh$, HNN-Snake, BNN-sin, HNN-sin. . . . 27
10 Mean of MSE of predictions for different modelling approaches with 2 SE (1 SE for BNN-$\tanh$) confidence intervals against time. 28
A.1 Radial distributions of data points for considered systems 34
B.1 Predicted (HNN) and True x_1 vs time for the test (left) and validation (right) initial conditions, ordered from top to bottom from system $N = 1$ to system $N = 5$. 35
C.1 Predicted (HNN) and True $\dot{x}_1$ vs time for the test (left) and validation (right) initial conditions, ordered from top to bottom from system $N = 1$ to system $N = 5$. 36
D.1 Predicted and True $\dot{x}_1$ vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-$\tanh$, HNN-$\tanh$, HNN-Snake, BNN-sin, HNN-sin. . . . 37

D.2 Predicted and True x_2 vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-$\tanh$, HNN-$\tanh$, HNN-Snake, BNN-$\sin$, HNN-$\sin$. . . . 38
D.3 Predicted and True $\dot{x}_2$ vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-$\tanh$, HNN-$\tanh$, HNN-Snake, BNN-$\sin$, HNN-$\sin$. . . . 39

Contents

1 Introduction **7**

2 Theoretical background **8**
2.1 Hamiltonian formalism 8
2.2 Fundamentals of neural networks 9

3 Literature Review **10**
3.1 Hamiltonian Neural Networks 10
3.2 Dynamical systems and coupled oscillators 11

4 Models and methodology **12**
4.1 Duffing oscillator 12
4.1.1 Undamped coupled Duffing oscillators in Hamiltonian formalism 13
4.2 System of Duffing oscillators with non-linear coupling and derivation of respective equations of motion from port-Hamiltonian formalism 15
4.3 Hamiltonian Neural Network approach 16
4.4 Port-Hamiltonian Neural Network approach 16
4.5 Baseline Neural Network approach 17

5 Datasets **18**
5.1 Unforced and undamped oscillators 19
5.2 Forced oscillators 20

6 Numerical experiments and results **21**
6.1 Unforced and undamped systems 21
6.1.1 Visual assessment of obtained predictions 21
6.1.2 Comparative analysis of performance between HNN and NN approaches 21
6.1.3 A closer look at the generalisation 23
6.2 Forced and damped system 25
6.2.1 Predictions obtained with different approaches 25
6.2.2 Comparative analysis between approaches 26

7 Discussion **28**
7.1 Limitations of our experimental settings 28
7.2 Summary of findings 28
7.3 Recommendations for future research 29

Appendix A Radial distributions of datapoints for the unforced and undamped systems **34**

Appendix B Graphs of 1-st oscillator coordinate predicted with HNN for the unforced and undamped systems **35**

Appendix C Graphs of 1-st oscillator velocity predicted with HNN for the unforced and undamped systems **36**

Appendix D Additional plots for coordinates and velocities of the first oscillator with different approaches for the forced and damped system **37**

I. Licence 40

1 Introduction

Dynamical systems can describe a breadth of phenomena, simpler examples are radioactive decay, populations of single species and RLC circuits. More complex phenomena include modelling economics, molecular dynamics, ecosystems, nonlinear optics and many more [1]. As such, modelling the behaviour of dynamical systems is the fundamental task for many disciplines of science. For many phenomena, creating a mathematical model from data can prove to be very difficult due to the complexity of the problem and lack of first-principle understanding of the phenomenon (e.g. see [2] and references therein). Neural Networks (NNs) are universal function approximators [3], which can fit complex patterns and emulate nonlinear dynamics by learning on data and as such are a promising alternative way to model dynamical systems. However, neural networks in their basic form lack physical constraints, and do not generalise well. Physics-guided NNs represent a class of approaches, that combines physical concepts with neural networks, sometimes adopting the deep learning settings (e.g. see [4] and references therein). It is worth mentioning that deep learning approaches have shown success in modelling fluid dynamics, molecular dynamics, materials science and quantum systems [5–7], but here we are considering somewhat smaller NNs that remain outside of the scope of DL.

In this thesis, we focus on two approaches, namely, Hamiltonian Neural Network (HNN) and port-Hamiltonian neural network (pHNN). These approaches introduce the intrinsic constraints of canonical equations of Hamiltonian formalism onto standard neural networks. The latter also allows for modelling of dissipative systems influenced by external forces. [8, 9]

Coupled oscillators are used to model several real-world cases. Even a coupled system with size two or three can be a valuable model for some problems (e.g. [10, 11]). To our best knowledge, the application of HNNs to coupled oscillators has not been investigated yet (e.g. see [9, 12]). For instance, in [12] the emphasis was on the conservative dynamics of systems such as the Lotka–Volterra predator–prey model, the elastic and double pendulum. A simple Duffing oscillator has been considered in [9]. We test whether the HNN and the pHNN are suitable approaches for modelling systems of coupled Duffing oscillators, and explore minor alterations to the pHNN network architecture by using different activation functions in one of the layers. The Hamiltonian approaches are also compared against baseline non-Hamiltonian NN approaches to determine whether the added constraints yield and improvement to resultant models. Numerical experiments are conducted with the HNN approach on systems of up to five coupled Duffing oscillators without forcing and damping and with pHNN approaches on a system of two coupled Duffing oscillators with non-zero forcing and damping terms, as initial investigation to assess the potential of tested approaches. The strength and the limitations of the HNN-based modelling approaches are highlighted, which grant insight into their practical viability and guide further refinement of the method.

2 Theoretical background

2.1 Hamiltonian formalism

The following short overview is based on [13]. Assuming that a given system is holonomic, and the forces are derived from potentials dependent on position and time only. Given k holonomic constraints, the Cartesian coordinates of constituent parts of the system can be expressed in terms of n equations of $N = 3n - k$ generalised coordinates q_i:

$$\begin{aligned} \boldsymbol{r}_1 &= \boldsymbol{r}_1(q_1, q_2, ..., q_N, t), \\ &\vdots \\ \boldsymbol{r}_n &= \boldsymbol{r}_n(q_1, q_2, ..., q_N, t). \end{aligned}$$

As such, the dynamics of the system is fully described by the generalised coordinates and the system is said to have N degrees of freedom. For systems in one-dimensional coordinates, there are only $N = n - k$ generalised coordinates, and the same number of generalised momenta, respectively:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}, \boldsymbol{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}$$

In the Hamiltonian formalism, the dynamics of the system with N degrees of freedom is given by $2N$ first-order differential equations:

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \ \dot{q}_i = \frac{\partial H}{\partial p_i}, \ i = 1, 2, ..., N \tag{1}$$

where the Hamiltonian H of the system is constructed using the Legendre transformation:

$$H(\boldsymbol{q}, \boldsymbol{p}, t) = \sum_{i=1}^{N} \dot{q}_i p_i - L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$$

and $L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = T - V$ is the Lagrangian of the system. T is the total kinetic energy of the system, and V is the total potential energy of the system. The generalised or conjugate momenta are defined as:

$$p_i = \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \dot{q}_i}.$$

The set of generalised coordinates and generalised momenta $(\boldsymbol{q}, \boldsymbol{p})$ are known as canonical coordinates and can be treated as a single column vector:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}.$$

By introducing the following:

$$\boldsymbol{J} \equiv \boldsymbol{J}_N = \begin{bmatrix} \mathbf{0} & \boldsymbol{I}_N \\ -\boldsymbol{I}_N & \mathbf{0} \end{bmatrix}, \ \nabla_{\boldsymbol{\eta}} H = \begin{bmatrix} \frac{\partial H}{\partial \eta_1} \\ \vdots \\ \frac{\partial H}{\partial \eta_{2N}} \end{bmatrix},$$

where $\boldsymbol{I}_N$ is the identity matrix of size N. Then (1) can be expressed as:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J} \nabla_{\boldsymbol{\eta}} H(\boldsymbol{\eta}, t). \tag{2}$$

The Hamiltonian allows to describe the time evolution of the system in $2N$-dimensional phase $\{q_i, p_i\}$ space by (2). Hence, the dynamics of the system, i.e. any point $\boldsymbol{\eta}_1$ on the unique path in phase space starting at a point with coordinates $\boldsymbol{\eta}_0$ can be found by integrating over time [8]:

$$\boldsymbol{\eta}_1 = \boldsymbol{\eta}_0 + \int_{t_0}^{t_1} \dot{\boldsymbol{\eta}} dt.$$

2.2 Fundamentals of neural networks

In this thesis we consider only a subset of neural networks known as multi-layer feed-forward neural networks (FFNN). They have a layered structure, where each layer consists of neurons, and each neuron receives the input from neurons in a layer directly before it, and its output propagates to neurons in the layer after it, with no interconnection between neurons in a given layer (see Fig. 1) [14]. Neural networks are universal function approximators [3], with many parameters in the form of weights w_{ji} and biases b_i for each layer, defining its connection to the layer before it. We denote the set of all parameters of a NN-based model as θ, and consequently $\{w_{ji}^k, b_i^k\} \in \theta$, where the index k represents the layer number. The output of i-th neuron in layer k, besides the input layer, is given by:

$$x_i^k = f\left(b_i^k + \sum_j w_{ji}^k x_j^{k-1}\right)$$

Where $f(x)$ is an activation function. [15] It is important to note, that $f(x)$ can also be a linear function, i.e $f(x) = x$, which is an usual choice for the output layer, as in our case.

The process of training neural network consists of optimising the parameters θ of a neural network $\boldsymbol{G}_\theta(\boldsymbol{x}_i)$ in such a way so that, given the input vectors $\boldsymbol{x}_i$, the output vectors $\boldsymbol{G}_\theta(\boldsymbol{x}_i) = \boldsymbol{y}_i$ match the desired output vectors $\boldsymbol{y}_i^d$ (or targets). Consequently, the training datasets consist of pairs of vectors $(\boldsymbol{x}_i, \boldsymbol{y}_i^d)$. To optimise the parameters of the neural network, a loss (or error) function L_θ has to be defined, such as the mean squared error (MSE):

$$L_\theta = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{y}_i^d - \boldsymbol{G}_\theta(\boldsymbol{x}_i))^2,$$

where M is the number of samples. The set of parameter values θ which minimises the error function can then be found by the so-called back-propagation, by means of algorithms such as Stochastic Gradient Descent or Adam, satisfying:

$$\theta = \arg\min_\theta L_\theta$$

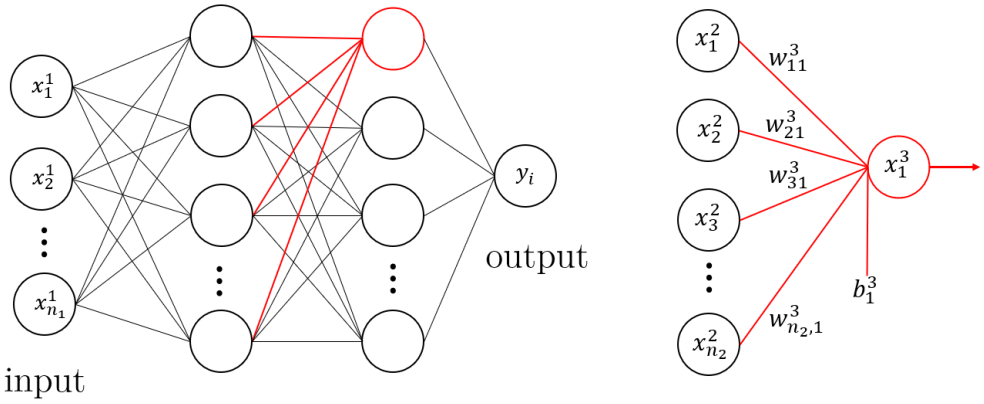

Figure 1. Structure of feedforward neural network with densely connected layers, n_i is the number of neurons in a i-th layer.

3 Literature Review

3.1 Hamiltonian Neural Networks

Hamiltonian Neural Networks, as considered in this thesis, were first proposed in [8] in 2019. In their method, a machine learning approach was used to parametrise a Hamiltonian of a system with a neural network and optimise it by learning on coordinates and

their time-derivatives from trajectories in the phase space of the system. The authors of the article analysed the performance of their method on simple mass-spring, pendulum and N-body systems and found that it reproduced the dynamics of the systems well. Their approach has an advantage over the basic FFNN approach, because when the model is used to parametrise the Hamiltonian it gains certain physical priors intrinsic to Hamiltonian formalism. Indeed, as the authors of [8] showed, the dynamics forecast by the FFNN models do not exhibit energy conservation and the predictions diverge from ground truth quickly. The HNN models performed much better, forecasting dynamics, which deviate from the ground truth less and model energy conservation well, as energy conservation follows from the time-independence of the Hamiltonian in the systems they consider. The authors also extended their approach by adding an autoencoder to their model, and using the latent space of the autoencoder as the input to the HNN. The model trained on pairs on sequential 16×16 images of a simulated pendulum, and was able to recover the Hamiltonian of the system from pixel data.

In [9] the approach was extended to Port-Hamiltonian formalism, which allows for energy dissipation and a control input in the system. The authors compared their extended HNN approach, namely the Port-Hamiltonian Neural networks (pHNN), against two other approaches, including one proposed in [8]. The comparison is made based on performance in modelling more complex dynamical systems, such as forced and damped mass-spring systems, a driven relativistic particle in double well potential, and Duffing equation in chaotic and non-chaotic regimes. They found that their pHNN approach outperforms the approach proposed in [8] for all mentioned systems, yielding good approximation to the true dynamics of the considered systems.

In [12] another extension to the HNN method described in [8] is proposed. It involves prepending a conventional neural network to the Hamiltonian neural network. The role of the prepended neural network is to transform non-canonical coordinates to canonical coordinates, which is useful when canonical coordinates cannot be easily inferred from data. The authors called their extension Generalised HNN (gHNN) approach, and showed that it outperforms the regular HNN approach for systems such as predator-prey population dynamics and an elastic pendulum, and generally recovers the dynamics of the systems well.

3.2 Dynamical systems and coupled oscillators

Dynamical systems have a wide range of applications. In cosmology they can be used to model the evolution of cosmological objects or the universe as a whole, which allows to test different models proposed by theoretical physicists [16–18]. Other applications include research in ecological systems, control theory, chemical reaction modelling, disease modelling, epidemiology and more [19].

Among dynamical systems, coupled oscillators (COs) have been paid remarkable attention. Because of their complex behaviour they allow for a better modelling of real-world cases. COs have been used to model the human vocal cords (e.g. see [11] and references therein) and for describing large scale brain activity and in particular its metastability [20]. Coupled oscillators model the strong coupling in graphene-based nanostructures [10]. Human walking can also be regarded as an oscillating system and the environment as a second oscillating system. Hence, the human and the environment together constitute a coupled oscillator system [21]. The small intestine can be modelled by a network of coupled oscillators [22]. Kuramoto used the theory of coupled oscillators to show turbulence in fluid dynamics [23]. COs were used to to analyze the sectoral synchronisation observed for the Japanese business cycle of an economic system made up of industry sectors and goods markets [24]. Modular robots have also been modelled by using COs [25].

4 Models and methodology

4.1 Duffing oscillator

The equation governing the dynamics of the Duffing oscillator is:

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t).$$

[26] Where $\delta, \alpha, \beta, \gamma, \omega$ are constants. We consider linear two-way coupling between oscillators, which for a system of two oscillators can be expressed as:

$$\begin{aligned}\ddot{x}_1 + \alpha_1 x_1 + \beta_1 x^3 &= \gamma_1\cos(\omega_2 t) + \varepsilon(x_2 - x_1)\\ \ddot{x}_2 + \alpha_2 x_2 + \beta_2 x_2^3 &= \gamma_2\cos(\omega_1 t) + \varepsilon(x_1 - x_2)\end{aligned}$$

Where ε is the coupling constant. Systems of N Duffing oscillators are used in our experiments, where each oscillator is linearly coupled with all other oscillators, with governing equations for i-th oscillator being:

$$\ddot{x}_i + \alpha_i x_i + \beta_i x_i^3 = \gamma\cos(\omega t) + \sum_{j=1}^{N}\varepsilon_{ij}(x_j - x_i) \tag{3}$$

Where ε_{ij} is a matrix representing the coupling coefficients between oscillators. We can rewrite (3) as a system of 1-st order differential equations for numerical integration:

$$\begin{aligned}\dot{x}_i &= y_i\\ \dot{y}_i = \ddot{x}_i &= -\alpha_i x_i - \beta_i x_i^3 + \gamma_i\cos(\omega_i t) + \sum_{j=1}^{N}\varepsilon_{ij}(x_j - x_i)\end{aligned} \tag{4}$$

4.1.1 Undamped coupled Duffing oscillators in Hamiltonian formalism

In this section we show how the equations of motions (4) can be obtained by using the Hamiltonian formalism. We also show that for this particular system, the pairs $(x_i, \dot{x}_i)$ are valid canonical coordinates and thus can be used as inputs to the HNN.

We follow the steps provided in [13, pp. 338] to construct the Hamiltonian. We define generalised momenta as $q_i = x_i$. Kinetic (T) and potential (V) energies, assuming that the system elements have masses m_i, are the following:

$$T = \sum_{i=1}^{N} \frac{m_i \dot{x}_i^2}{2} = \sum_{i=1}^{N} \frac{m_i \dot{q}_i^2}{2},$$

$$V = \sum_{i=1}^{N} \left[\frac{\alpha_i'}{2} q_i^2 + \frac{\beta_i'}{2} q_i^4 - \gamma_i' q_i \cos(\omega_i t) \right] + \frac{1}{2} \sum_{i=1}^{N} \sum_{j>i}^{N} \left[\varepsilon_{ij}' \left(q_j - q_i \right)^2 \right].$$

Where ε_{ij}' represents the coupling coefficients between oscillators and is also defined to be symmetric for convenience in this derivation. The Lagrangian becomes:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = T - V =$$

$$= \sum_{i=1}^{N} \left[\frac{m_i \dot{q_i}^2}{2} - \frac{\alpha_i'}{2} q_i^2 - \frac{\beta_i'}{2} q_i^4 + \gamma_i' q_i \cos(\omega_i t) \right] - \frac{1}{2} \sum_{i=1}^{N} \sum_{j>i}^{N} \left[\varepsilon_{ij}' \left(q_j - q_i \right)^2 \right].$$

This yields the following canonical momenta:

$$p_i = \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \dot{q}_i} = m_i \dot{q}_i$$

and the Hamiltonain is obtained as:

$$H(\boldsymbol{q}, \boldsymbol{p}, t) = \sum_{i=1}^{N} \dot{q}_i p_i - L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) =$$

$$= \sum_{i=1}^{N} [m_i \dot{q}_i^2] - \sum_{i=1}^{N} \left[\frac{m_i \dot{q_i}^2}{2} - \frac{\alpha_i'}{2} q_i^2 - \frac{\beta_i'}{2} q_i^4 + \gamma_i' q_i \cos(\omega_i t) \right] + \frac{1}{2} \sum_{i=1}^{N} \sum_{j>i}^{N} \left[\varepsilon_{ij}' \left(q_j - q_i \right)^2 \right] =$$

$$= \sum_{i=1}^{N} \left[\frac{p_i^2}{2m_i} + \frac{\alpha_i'}{2} q_i^2 + \frac{\beta_i'}{2} q_i^4 - \gamma_i q_i \cos(\omega_i t) \right] + \frac{1}{2} \sum_{i=1}^{N} \sum_{j>i}^{N} \left[\varepsilon_{ij}' \left(q_j - q_i \right)^2 \right] = T + V \tag{5}$$

where we made the substitution $m_i \dot{q}_i = p_i$. The partial derivative of the term in the Hamiltonian that represents the potential of couplings with respect to the canonical

coordinate q_k is:

$$\frac{\partial}{\partial q_k}\left[\frac{1}{2}\sum_{i=1}^{N}\sum_{j>i}^{N}\left[\varepsilon'_{ij}\left(q_j-q_i\right)^2\right]\right]=\frac{1}{2}\sum_{j>k}^{N}\varepsilon'_{kj}\frac{\partial}{\partial q_k}(q_j-q_k)^2+\frac{1}{2}\sum_{i<k}^{N}\varepsilon'_{ik}\frac{\partial}{\partial q_k}(q_k-q_i)^2=$$
$$=\sum_{j>k}^{N}\varepsilon'_{jk}(q_k-q_j)+\sum_{i<k}^{N}\varepsilon'_{ik}(q_k-q_i)=\sum_{i=1,i\neq k}^{N}\varepsilon'_{ik}(q_k-q_i)=-\sum_{i=1,}^{N}\varepsilon'_{ik}(q_i-q_k)$$

Where we used $\frac{\partial}{\partial x}(x-y)^2=\frac{\partial}{\partial x}(y-x)^2=2(x-y)$, and the property of the defined matrix $\varepsilon'_{ij}=\varepsilon'_{ji}$. Hence, the canonical equations are:

$$\begin{aligned}\dot{q}_i&=\frac{\partial H(\boldsymbol{q},\boldsymbol{p},t)}{\partial p_i}=\frac{\dot{p}_i}{m_i}\\ \dot{p}_i&=-\frac{\partial H(\boldsymbol{q},\boldsymbol{p},t)}{\partial q_i}=-\alpha'_i q_i-\beta'_i q_i^3+\gamma'_i\cos(\omega_i)+\sum_{j=1,}^{N}\varepsilon'_{ij}(q_j-q_i)=\ddot{q}_i m_i\end{aligned}\tag{6}$$

From (6) we get the equations of motion for the considered system of undamped coupled Duffing oscillators (see (4)):

$$\begin{aligned}\dot{q}_i&=\frac{\dot{p}_i}{m_i}=\frac{m_i\dot{q}_i}{m_i}=\dot{q}_i=\dot{x}_i\\ \ddot{q}_i&=-\alpha_i q_i-\beta_i q_i^3+\gamma_i\cos(\omega_i)+\sum_{j=1}^{N}\left[\varepsilon_{ij}(q_j-q_i)\right]=\ddot{x}_i\end{aligned}\tag{7}$$

where $\alpha_i=\alpha'_i/m_i$, $\beta_i=\beta'/m_i$, $\gamma_i=\gamma'_i/m_i$, $\varepsilon_{ij}=m_i^{-1}\varepsilon'_{ij}$. Let us now rewrite the Hamiltonian in (5) by substituting $\alpha'_i=m_i\alpha_i$, $\beta'_i=m_i\beta_i$, $\gamma'_i=\gamma_i m_i$ and $\varepsilon'_{ij}=m_i\varepsilon_{ij}$, we obtain:

$$H(\boldsymbol{q},\boldsymbol{p},t)=$$
$$=\sum_{i=1}^{N}\left[\frac{p_i^2}{2m_i}+\frac{m_i\alpha_i}{2}q_i^2+\frac{m_i\beta_i}{4}q_i^4-m_i\gamma_i q_i\cos(\omega_i t)\right]+\frac{1}{2}\sum_{i=1}^{N}\sum_{j>i}^{N}\left[m_i\varepsilon_{ij}\left(q_j-q_i\right)^2\right].\tag{8}$$

As we have shown, the Hamiltonian in (8) produces the equations of motion for coupled undamped Duffing oscillators with parameters $\alpha_i,\beta_i,\gamma_i,\omega_i,\varepsilon_{ij}$ in (4). Therefore, parameters m_i can be freely chosen (except $m_i=0$) without affecting the resultant equations of motion (7). For the sake of convenience, we set $m_i=1$, i.e. $q_i=x_i$ and $p_i=\dot{x}_i$. For these canonical coordinates, the Hamiltonian that produces exactly the equations of motion (4) is therefore:

$$H(\boldsymbol{q},\boldsymbol{p},t)=\sum_{i=1}^{N}\left[\frac{p_i^2}{2}+\frac{\alpha_i}{2}q_i^2+\frac{\beta_i}{4}q_i^4-\gamma_i q_i\cos(\omega_i t)\right]+\frac{1}{2}\sum_{i=1}^{N}\sum_{j>i}^{N}\left[\varepsilon_{ij}\left(q_j-q_i\right)^2\right].$$

4.2 System of Duffing oscillators with non-linear coupling and derivation of respective equations of motion from port-Hamiltonian formalism

We consider a system of coupled Duffing oscillators that exhibits periodic trajectories for known parameter values [27]. The equations of motion for the system take the following form:

$$\begin{aligned} \ddot{x}_1 &= -2\alpha_1 x - 4\beta_1 x^3 - 2\varepsilon x_1 x_2^2 - \delta_1 \dot{x}_1 + \gamma_1 \cos\omega_1 t \\ \ddot{x}_2 &= -2\alpha_2 x - 4\beta_2 x^3 - 2\varepsilon x_2 x_1^2 - \delta_2 \dot{x}_2 + \gamma_2 \cos\omega_2 t. \end{aligned} \tag{9}$$

Hamilton's equations in port-Hamiltonian formalism can be expressed as follows:

$$\dot{\boldsymbol{\eta}} = [\boldsymbol{J} + \boldsymbol{D}(\boldsymbol{q})]\nabla_{\boldsymbol{\eta}} H(\boldsymbol{\eta}) + \boldsymbol{G}(\boldsymbol{q})u, \tag{10}$$

where $\boldsymbol{D}(\boldsymbol{q})$ is an $2N \times 2N$ matrix representing damping, $\boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{g}(\boldsymbol{q})\end{bmatrix}^T$, u is a temporal control input, and H is the Hamiltonian [9, 28]. Eq. (10) can be specialised for coupled Duffing oscillators with forcing and damping terms in (9), by defining:

$$\begin{aligned} \boldsymbol{D}(\boldsymbol{q}) &= \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{\delta}\end{bmatrix} \\ \boldsymbol{F}(t) &\equiv \begin{bmatrix}\gamma_1 \cos\omega_1 t \\ \gamma_2 \cos\omega_2 t\end{bmatrix} \\ \boldsymbol{G}(\boldsymbol{q})u &\equiv \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{F}(t)\end{bmatrix} \\ H(\boldsymbol{q},\boldsymbol{p}) = T + V &= \sum_{i=1}^{2}\left[\frac{p_i^2}{2} + \alpha_i q_i^2 + \beta_i q_i^4\right] + \varepsilon q_1^2 q_2^2, \end{aligned} \tag{11}$$

where:

$$\begin{aligned} \boldsymbol{\delta} &= \begin{bmatrix}\delta_1 & 0 \\ 0 & \delta_2\end{bmatrix} \\ T &= \sum_{i=1}^{2}\frac{p_i^2}{2} \\ V &= \sum_{i=1}^{2}\left[\alpha_i q_i^2 + \beta_i q_i^4\right] + \varepsilon q_1^2 q_2^2 \end{aligned}$$

Substituting (11) into (10) results in the following equations of motion:

$$\dot{\boldsymbol{\eta}} \equiv \begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & -\boldsymbol{\delta}\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q_i} \\ \frac{\partial H}{\partial p_i}\end{bmatrix} + \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{F}(t)\end{bmatrix},\ i = 1, 2 \tag{12}$$

which yields:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial H}{\partial p_1} \\ \frac{\partial H}{\partial p_2} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \tag{13}$$

and:

$$\begin{aligned} \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} &= - \begin{bmatrix} \frac{\partial H}{\partial q_1} \\ \frac{\partial H}{\partial q_2} \end{bmatrix} - \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial p_1} \\ \frac{\partial H}{\partial p_2} \end{bmatrix} + \begin{bmatrix} \gamma_1 \cos(\omega_1 t) \\ \gamma_2 \cos(\omega_2 t) \end{bmatrix} = \\ &= \begin{bmatrix} -2\alpha_1 q_1 - 4\beta_1 q_1 - 2\varepsilon q_1 q_2^2 - \delta_1 p_1 + \gamma_1 \cos(\omega_1 t) \\ -2\alpha_2 q_2 - 4\beta_2 q_2 - 2\varepsilon q_2 q_1^2 - \delta_2 p_2 + \gamma_2 \cos(\omega_2 t) \end{bmatrix}. \end{aligned} \tag{14}$$

Assuming $\boldsymbol{x} = \boldsymbol{q}$ and $\dot{\boldsymbol{x}} = \boldsymbol{p}$, one can notice that eq. (14), derived from the equations expressing the port-Hamiltonian formalism (10), have the same form as (9). As we have shown, (12) can be used to describe the dynamics of the system of damped and forced coupled Duffing oscillators and thus we can apply the port-Hamiltonian neural network approach to this system.

4.3 Hamiltonian Neural Network approach

In the HNN approach, a neural network is used to approximate the Hamiltonian of the system. This means using the Neural Network instead of the Hamiltonian and obtaining the time derivative $\dot{\boldsymbol{\eta}}$ with canonical equations according to the Hamiltonian formalism (1), see Fig. 2. [8] In our experiments we use the MSE loss that can be expressed as:

$$L_\theta = \frac{1}{M} \sum_{i=1}^{M} \left[\dot{\boldsymbol{\eta}}_i^d - \boldsymbol{J} \nabla_{\boldsymbol{\eta}} H_\theta(\boldsymbol{\eta}_i) \right]^2$$

where $\dot{\boldsymbol{\eta}}_i^d$ are the desired impulse and momenta time derivatives, M is the number of samples, $H_\theta \equiv H_\theta(\boldsymbol{\eta})$ is the neural network with weights and biases collected in θ.

4.4 Port-Hamiltonian Neural Network approach

To model the Duffing oscillator with a non-zero damping and forcing terms, we utilise the Port-Hamiltonian Neural Network (pHNN) approach introduced in [9]. The port-Hamiltonian model consists of a regular HNN which fits the Hamiltonian in (12). To that another neural network is added, which takes time t as a single input, and outputs a vector of time-dependant forcing terms $\boldsymbol{F}_\theta(t)$ from (12), and a set of trainable parameters $\delta_{ij} \in \theta$ fitting the damping coefficients in (12), see Fig. 3. Regularisation parameters are also added to the loss function of the network, which encourage the model to minimise the magnitude of forcing and damping terms to prevent the model from learning spurious

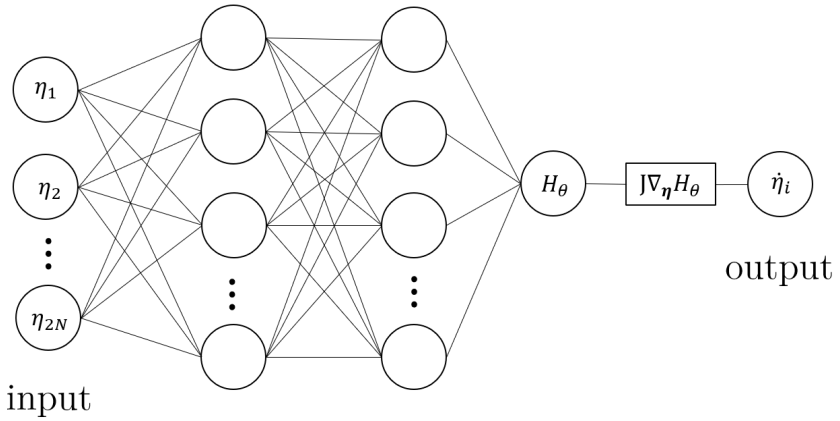

Figure 2. Structure of a Hamiltonian neural network

forces and damping parameters. The loss function is expressed as:

$$L_\theta = \frac{1}{M}\sum_{i=1}^{M}\left[\left(\dot{\boldsymbol{\eta}}_i^d - \left(\boldsymbol{J}_\delta \nabla_{\boldsymbol{\eta}} H_\theta(\boldsymbol{\eta}_i) + \begin{bmatrix}\boldsymbol{0} & \boldsymbol{F}_\theta(t_i)\end{bmatrix}^T\right)\right)^2 + \lambda_F \|\boldsymbol{F}_\theta(t_i)\|_1\right] + \lambda_\delta \sum_{i=1}^{N}\sum_{j=1}^{N}|\delta_{ij}|$$

where M is the number of samples, λ_F and λ_δ are regularisation parameters. [9] And

$$\boldsymbol{J}_\delta = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & -\boldsymbol{\delta}\end{bmatrix},$$

where $\boldsymbol{\delta}$ is initialised as a square zero matrix.

4.5 Baseline Neural Network approach

We compare the performance of the proposed HNN and pHNN approaches against a baseline neural network approach (BNN). In the baseline approach a neural network is used to directly predict time-derivatives of the canonical coordinates of the system, given the state of the system as input. For the forced system, where the equations of motion include time explicitly, time is also added as an input. The MSE loss function we use is defined as:

$$L_\theta = \frac{1}{M}\sum_{i=1}^{M}\left[\dot{\boldsymbol{\eta}}_i{}^d - \boldsymbol{G}_\theta(\boldsymbol{\eta}_i, t_i)\right]^2$$

Where $\boldsymbol{G}_\theta$ is the output vector of the network and M is the number of samples. The structure of this approach is depicted in Figure 4. We implement all the approaches using the `pytorch` Python package [29]. For the initialisation for weight and biases in all models the default settings are used.

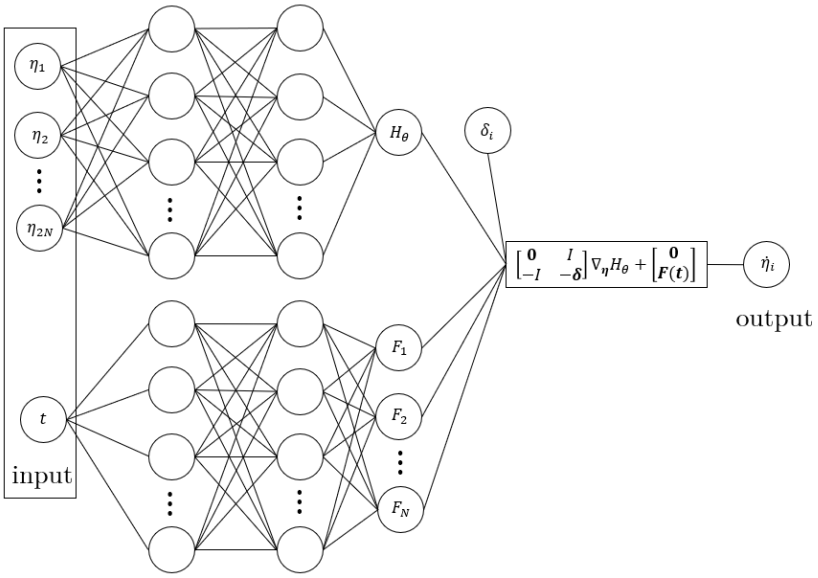

Figure 3. Structure of a port-Hamiltonian neural network

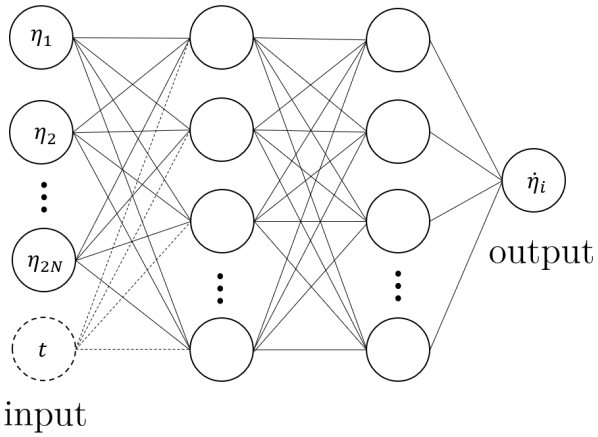

Figure 4. Structure of a Baseline neural network

5 Datasets

All datasets are generated by numerical integration of respective ODEs modelling the considered systems in Python. We utilise the `solve_ivp` function from `scipy` package with fifth-order Implicit Runge-Kutta method of the Radau IIA family as solver [30].

The datasets consist of pairs of vectors $(\boldsymbol{\eta}_i, \dot{\boldsymbol{\eta}}_i^d)$ at the time t_i in the form:

$$\boldsymbol{\eta}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \dot{\boldsymbol{x}}_i \end{bmatrix}, \quad \dot{\boldsymbol{\eta}}_i^d = \begin{bmatrix} \dot{\boldsymbol{x}}_i^d \\ \ddot{\boldsymbol{x}}_i^d \end{bmatrix}$$

where $\boldsymbol{\eta}_i$ are the inputs and $\dot{\boldsymbol{\eta}}_i^d$ are the desired (or true) outputs, both of size $2N$ for an N-degree-of-freedom system. We take the coordinates $\boldsymbol{x}_i$ and their derivatives $\dot{\boldsymbol{x}}_i$ from the numerical solutions (i.e trajectories in the phase space), and calculate the second-order derivatives using respective equations of motion for a given system.

5.1 Unforced and undamped oscillators

The equations of motion for the systems of undamped oscillators were introduced in section 4.1, we recall them here, setting the forcing terms $\gamma_i = 0$:

$$\ddot{x}_i + \alpha_i x_i + \beta_i x_i^3 = \sum_{j=1}^{N} \varepsilon_{ij}(x_j - x_i)$$

We consider systems of N coupled unforced and undamped oscillators, with $N = 2, \ldots, 5$, and a single unforced and undamped Duffing oscillator ($N = 1$). Being N the number of oscillators, it also represents the number of degrees of freedom the system. The system oscillators are denoted by Oi, meaning system with N oscillators consists of oscillators Oi, $i = 1, 2, \ldots N$. We set all values $\varepsilon_{ij} = 1$, other parameters and initial conditions (IC) for each oscillator are presented in Table 1.

For each system the training dataset is generated by varying the initial conditions x_0 presented in Table 1 and generating trajectories in the $2N$-dimensional phase space of the system. The training dataset consists of 1 trajectory generated with the initial conditions without variation, and 2^N trajectories generated for each combination of altered initial conditions x'_{0i}, which are changed as follows:

$$x'_{0i} = x_{0i} \pm 0.2, \ i = 1, ..., N.$$

The testing initial conditions for are obtained with following alteration:

$$\dot{x}'_{0i} = \dot{x}_{0i} + 0.1 \cdot (-1)^{i-1}, \ i = 1, ..., N.$$

And the validation initial conditions are:

$$\begin{aligned} x'_{0i} &= 1.3 x_{0i} \\ \dot{x}'_{0i} &= \dot{x}_{0i} + 0.2 \cdot (-1)^{i-1}, \ i = 1, ..., N. \end{aligned}$$

Testing and validation initial conditions allow to evaluate how the models generalise to different unseen initial conditions. The generated training trajectories span $t \in [0, 50]$,

and the data points are separated temporally by $dt = 0.05$, yielding $(1 + 2^N) \cdot 1001$ data points in the training set.

The choice of the base parameters and initial conditions was motivated on having a good variety of dynamics of oscillators. We include $\beta < 0$ and $\beta > 0$ nonlinearity values corresponding to softening and hardening springs, high and low values for stiffness, as well as small and large initial displacements. We also considered the radial distributions of the points in trajectories that result form each set of initial conditions, to provide some consistency in the results between systems. Firstly, we ensured that the distributions of points resulting from the testing initial conditions are roughly centred around the same point as the training data. Secondly, we ensured that the distributions for the validation initial conditions do not span far outside those for the training data, while still being significantly different from the training and testing distributions. The plots of distributions and a brief overview is provided in Appendix A.

Oscillator	α	β	x_{0i}	$\dot{x}_{0i}$
O_1	2	-0.4	0.8	0
O_2	0.7	3	-0.4	0
O_3	1	-0.5	0.2	0
O_4	0.3	4	-1	0
O_5	3	-0.4	-0.7	0

Table 1. Base parameters for Duffing oscillators

5.2 Forced oscillators

We recall the equations of a system of forced and damped oscillators introduced in section 4.2:

$$\ddot{x}_1 = -2\alpha_1 x_1 - 4\beta_1 x_1^3 - 2\varepsilon x_1 x_2^2 - \delta_1 \dot{x_1} + \gamma_1 \cos \omega_1 t$$
$$\ddot{x}_2 = -2\alpha_2 x_2 - 4\beta_2 x_2^3 - 2\varepsilon x_2 x_1^2 - \delta_2 \dot{x_2} + \gamma_2 \cos \omega_2 t$$

with parameters $\alpha_1 = 0.005$, $\alpha_2 = 0.01$, $\beta_1 = \beta_2 = 10$, $\varepsilon = 0.5$, $\delta_1 = \delta_2 = 3\alpha_1^{1/2}$, $\omega_1 = 1, \omega_2 = 0.73, \gamma_1 = \gamma_2 = 0.2$, which result in periodic trajectories [27]. The training dataset consists of data points from trajectories integrated from starting conditions $\boldsymbol{\eta}_{01}$ through $\boldsymbol{\eta}_{05}$ in Table 2 for $t \in [0, 50]$, $dt = 0.05$. This results in a total of 5005 data points. To test the accuracy of the models we use testing initial conditions $\boldsymbol{\eta}_{0\text{test}}$, and generate trajectories over a longer timespan $t \in [0, 100]$, $dt = 0.1$, when compared to the training dataset. This allows to evaluate how the models generalise to slightly different initial conditions and extrapolate in time.

	x_1	x_2	$\dot{x}_1$	$\dot{x}_2$
$\boldsymbol{\eta}_{01}$	0.3	0.3	0	0
$\boldsymbol{\eta}_{02}$	0.4	0.2	0	0
$\boldsymbol{\eta}_{03}$	0.4	0.4	0	0
$\boldsymbol{\eta}_{04}$	0.2	0.4	0	0
$\boldsymbol{\eta}_{05}$	0.2	0.2	0	0
$\boldsymbol{\eta}_{0\text{test}}$	0.25	0.25	0	0

Table 2. Initial conditions for forced Duffing oscillators

6 Numerical experiments and results

6.1 Unforced and undamped systems

For the unforced systems we use a HNN with 2 hidden layers with $\tanh$ activation functions. The input layer has size $2N$, where N is the number of degrees of freedom in the system, while the output layer consists of a single neuron corresponding to the value of the Hamiltonian. We set the size of both hidden layers to 128 per degree of freedom. For training we use the Adam optimiser with batches of size 128. We train the model for 20 epochs with learning rate equal to 10^{-3} and 40 epochs with learning rate of 10^{-4}. A comparison with baseline models is also made, BNNs have the same composition of hidden layers and are trained with the same procedure as the HNNs for the corresponding systems.

6.1.1 Visual assessment of obtained predictions

In Table 3 we provided brief granular comments on the predictions obtained with HNN approach. For systems $N = 2$ with testing initial conditions, and $N = 5$ with testing and validation initial conditions, poorer results were observed, with accuracy decreasing over time. Overall, the obtained predictions were accurate and the models generalised well to most novel initial conditions. The plots of predictions for x_1 and $\dot{x}_1$ are presented in Appendix B, with the plots for other coordinates being left out for brevity.

6.1.2 Comparative analysis of performance between HNN and NN approaches

To analyse the difference in the performance between HNN and NN approaches, we train 84 HNN and 84 NN models for each system and obtain the predicted trajectories for the testing and validation initial conditions. The mean squared errors of predictions

N	Testing Prediction	Validation prediction
1	Good with increasing error in frequency	Good with increasing error in frequency
2	Very good	Good decreasing to poor
3	Very good	Good
4	Good	Good with inaccuracies in fine features
5	Very good, quickly decreasing to very poor after $t = 25$	Good, quickly decreasing to poor

Table 3. Comments on the predictions obtained with HNN models for time-evolution of coordinate x_1

are summed over the time dimension, and the average and standard error (SE) of these sums are calculated for the 84 models trained for given system and model type. The sum of MSE over time (total MSE) for given initial conditions serves as a metric of the inaccuracy of the predicted trajectory. While the averages and standard errors allow to find statistically significant differences between the BNN and HNN approaches. The

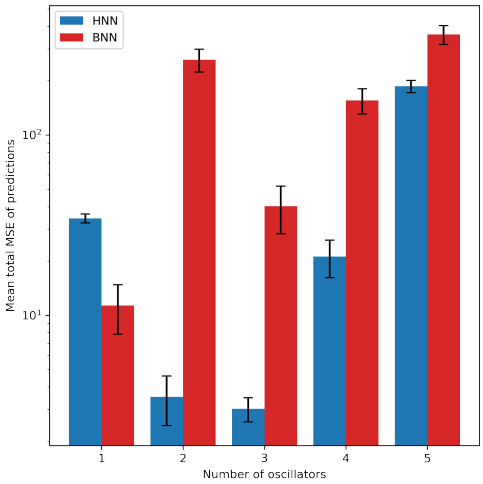

Figure 5. Mean total errors for testing initial conditions

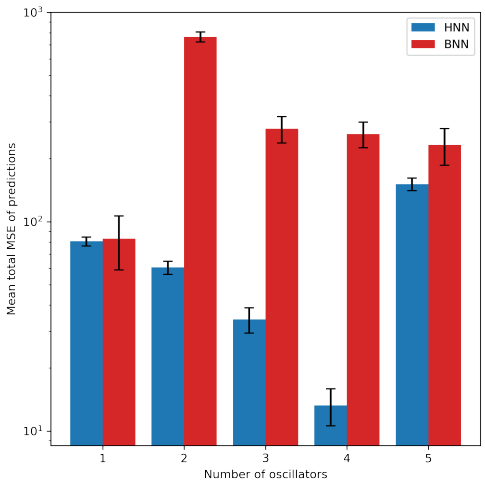

Figure 6. Mean total errors for validation initial conditions

results can be observed in Fig. 5, 6, with error bars corresponding to 2 SE, i.e 95% confidence interval, and in Table 4. The HNN models show a much lower SE of the the total MSE for all systems, indicating improved consistency of resultant models over the

	Testing ICs		Validation ICs	
N	HNN	BNN	HNN	BNN
1	34.44 ± 2	11.30 ± 3.46	80.58 ± 3.92	82.79 ± 23.88
2	3.52 ± 1.08	260.74 ± 37.66	60.42 ± 4.36	763.94 ± 42.75
3	3.03 ± 0.46	40.16 ± 11.84	34.10 ± 4.71	277.56 ± 40.19
4	21.11 ± 4.96	155.17 ± 24.83	13.26 ± 2.67	262.29 ± 36.64
5	186.09 ± 14.46	360.81 ± 43.19	151.09 ± 10.27	232.45 ± 46.60

Table 4. Mean total errors for different systems, models and initial conditions, presented with 2 SE uncertainty.

BNN approach. Moreover, for all $N > 1$ systems, and both testing and validation initial conditions, the HNN models show a statistically significant improvement to the accuracy of the predictions over the baseline approaches. As such, our analysis indicates that the constraints from the Hamiltonian formalism provide an improvement to the accuracy and consistency of the models.

6.1.3 A closer look at the generalisation

To further inspect how the model optimises for the training data we inspect the model with a single oscillator. For this system, we sample the MSE of the model predictions for time derivatives on 101×101 grid in its 2-dimensional phase space, where $x_1, \dot{x}_1 \in [-1.5, 1.5]$, and plot the loss as a heat map along with the trajectories from the training dataset in Fig. 7. We also explore for which initial conditions the model produces more accurate predictions. Hence, we calculate the predicted and true trajectories for $t \in [0, 10]$ for each set of initial conditions on a 41×41 grid of phase space coordinates, where $x_1, \dot{x_1} \in [-1.5, 1.5]$. Then, the average MSE of the predicted trajectory is evaluated for each sampled pair of initial conditions, and the results are plotted as a heat map. We also plot training trajectories and the true and predicted trajectories at selected points on the grid for reference. The true and predicted trajectories are plotted for $t \in [0, 2.9]$, as it can be seen in Fig. 8.

Looking at Fig.7, we see that the MSE valleys (regions with low MSE values) conform to the trajectories from the training set, meaning that the model does not generalise to the whole phase space of the system, but only in the space around those trajectories. Moreover, it can be observed, that the model does not optimise for all the data in the training dataset equally, as the MSE valleys conform to the training trajectories only in some places and are not continuous. In other places, loss valleys reside in-between training paths or form other artefacts. However, the error is still generally lower in the

region bound by the outer path, indicating that the model generalised better in that region.

Figure 8 implies that the model produces the most accurate predictions for ICs in regions close to the training trajectories. These regions conform to the shape shape of training trajectories much more closely, displaying less artefacts, if compared to the errors for time derivatives in Fig.7.

The shape of the sampled trajectories conform to the shape of true trajectories closely. However, a discrepancy in the length of trajectory can be observed for initial conditions that are further to the points of the training set. The former indicates inaccurate predictions for the frequency of oscillations.

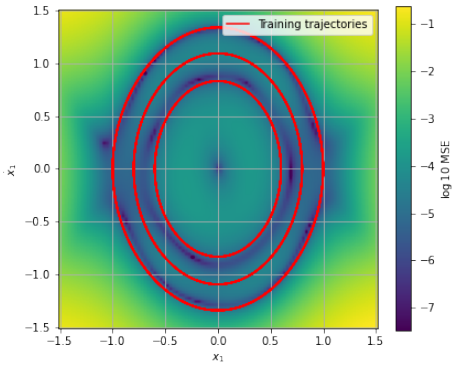

Figure 7. Heat map of MSE for predicted derivatives of canonical coordinates at each for points on a grid in phase space ($N = 1$)

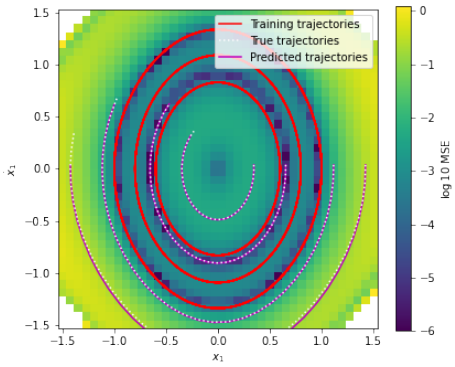

Figure 8. Heat map of MSE for predicted trajectories in timespan of $t \in [0, 10]$ with each points on a grid in as initial conditions, along with sampled trajectories for $t \in [0, 2.9]$ ($N = 1$)

6.2 Forced and damped system

For the forced and damped system, we evaluate how well the pHNN models predict the periodic behaviour of the system in (15), and compare the results against those obtained with baseline, non-Hamiltonian neural-network models, denoted as BNN. We also consider a longer timespan $t \in [0, 50]$ and $t \in [50.05, 100]$, compared to the reference article that introduced the technique, where $t \in [0, 10]$ was considered for a forced Duffing oscillator [9].

Moreover, we explore how the models extrapolate in time, which was not considered in the article that introduced the technique ([9]). Thusly, our training dataset spans $t \in [0, 50]$, but we evaluate and analyse the predictions for $t \in [0, 100]$. Extrapolation is of particular interest, since it has been shown in [31] that neural networks struggle with extrapolating periodic functions, and the forcing terms in the equation of motion of the system are periodic.

For the pHNN models we use $\tanh$ activation functions for every hidden layer besides the first hidden layer of the neural network that predicts the time-dependent forcing terms, where we propose approaches modified with different activation functions. For our experiments, we use 2 hidden layers with 512 neurons in the time-independent part of the models, and 2 hidden layers with 256 neurons each in the time-dependent part of the models. The values regularisation parameters $\lambda_F = \lambda_\delta = 10^{-5}$ were decided upon manual by fine-tuning, and training was done with Adam optimiser for 10 epochs with learning rate 10^{-3} and 30 epochs with learning rate 10^{-4}. For the BNN we use 768 neurons, which is equal to the total number of neurons in pHNN models, and the training procedure used is the same as for the pHNN models. The activation function for the BNN approaches is varied in the first hidden layer.

For the activation function of the first hidden layer of the time-dependant neural network in the pHNN approach, we explore $\tanh$, $\sin$, and the Snake activation function proposed in [31]. The latter two functions provide an inductive bias for the periodic forcing terms of considered oscillators. While $\tanh$ was used in articles introducing hamiltonian neural network approaches, including the pHNN approach [8, 9, 12]. The Snake activation function is given by:

$$y_i = \text{Snake}_a(x_i) = x_i + \frac{1}{a_i} \sin^2(a_i x_i)$$

Where a_i are trainable parameters and are initialised as $a_i = 1$. For the baseline approach we experiment with $\tanh$ and $\sin$ activation functions in the first hidden layer.

6.2.1 Predictions obtained with different approaches

The predictions for the time-evolution of coordinate x_1 obtained with the models we tested can be observed in Figure 9, and graphs the predictions for coordinates and velocities $\dot{x}_1$, x_2, $\dot{x}_2$ are included Appendix C.

Based on the visual assessment of the plots, the HNN models with $\tanh$ and Snake activation functions, and BNN with $\tanh$ activation, provide very inaccurate predictions over the whole timespan. We also do not observe an improvement with the Snake activation function when compared $\tanh$ in the pHNN models. Both BNN and HNN models with $\sin$ activations yield very good predictions in the timespan $t \in [0, 50]$, while for $t \in [50.05, 100]$ the models produce inaccurate results, which indicates overfitting. These models also generalised successfully to unseen initial conditions, modelling trajectories not included in the training dataset well in the first half of considered timespan.

6.2.2 Comparative analysis between approaches

We statistically analyse the performance of pHNN and BNN models with different activation functions. For that, we train each model for 100 times and find the average prediction MSE an the SE of prediction MSE at each time step for each approach for testing initial conditions. The results can be seen in Figure 10 with the shaded regions being confidence intervals of 2 SE for all approaches besides BNN with $\tanh$ activation, where we present confidence interval of 1 SE for visual clarity.

Analysing Figure. 10, in the unobserved timespan $t \in [50.05, 100]$ all HNN approaches converge to yield similarly poor performance, while the BNN approaches show even larger errors. The BNN-$\tanh$ approach showed the overall worst performance yielding a large SE of prediction errors and the highest average error for most of the timespan $t \in [0, 100]$.

For the timespan included in the training data $t \in [0, 50]$, we see similarly high average MSE for the HNN approaches with $\tanh$ and Snake activation function, further indicating, that the Snake function does not yield and improvement in our case. We observe the lowest average MSE with the HNN and BNN models that include the sin activation function, which indicates that the inductive bias from the $\sin$ activation function has a positive effect on the accuracy of predictions. From the two approaches, the HNN approach yields a statistically significant improvement in accuracy over the baseline approach for most of the timespan present in the training data, indicating and that the added constraints from the Hamiltonian formalism further contribute to the accuracy of the models.

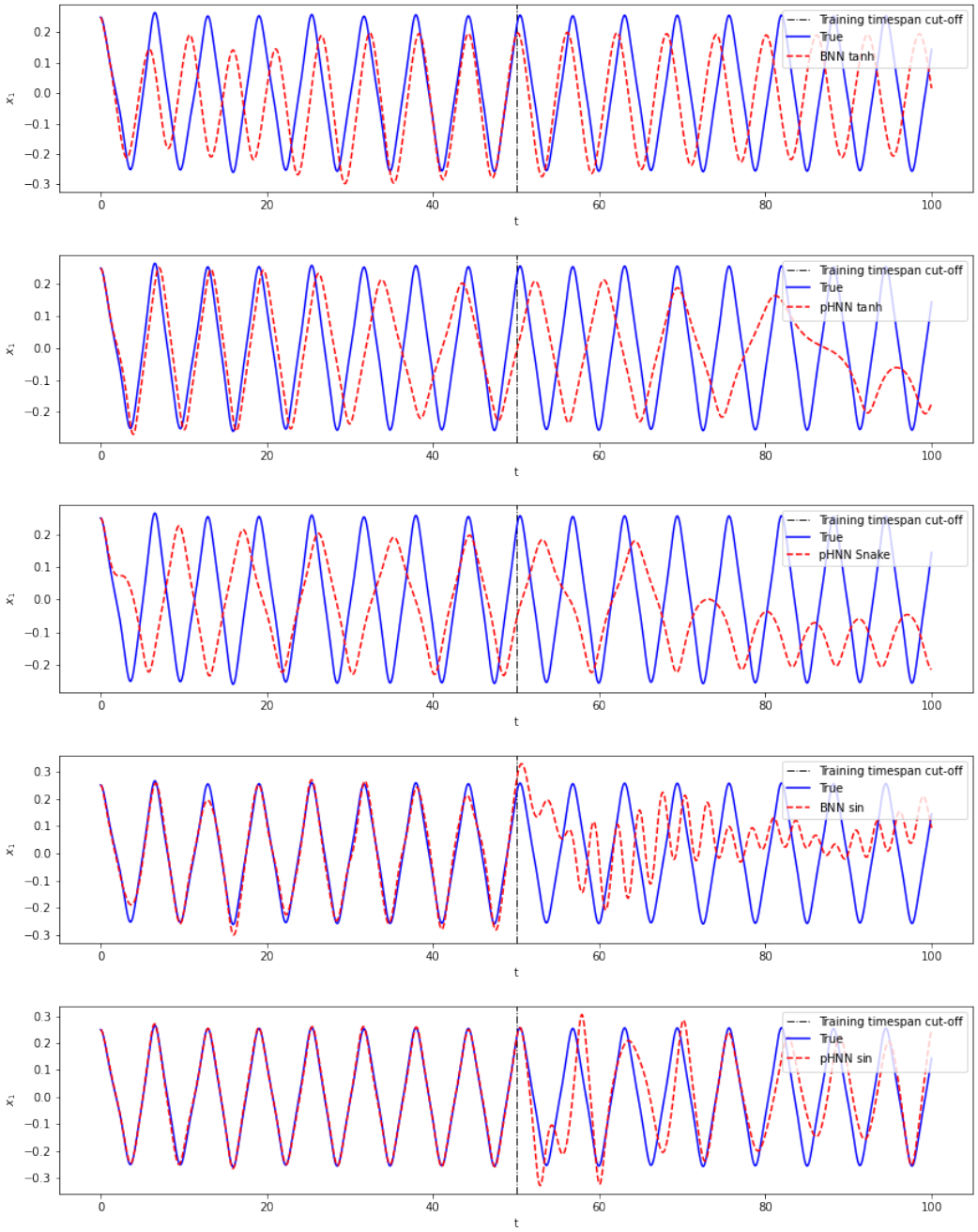

Figure 9. Predicted and True x_1 vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-tanh, HNN-tanh, HNN-Snake, BNN-sin, HNN-sin.

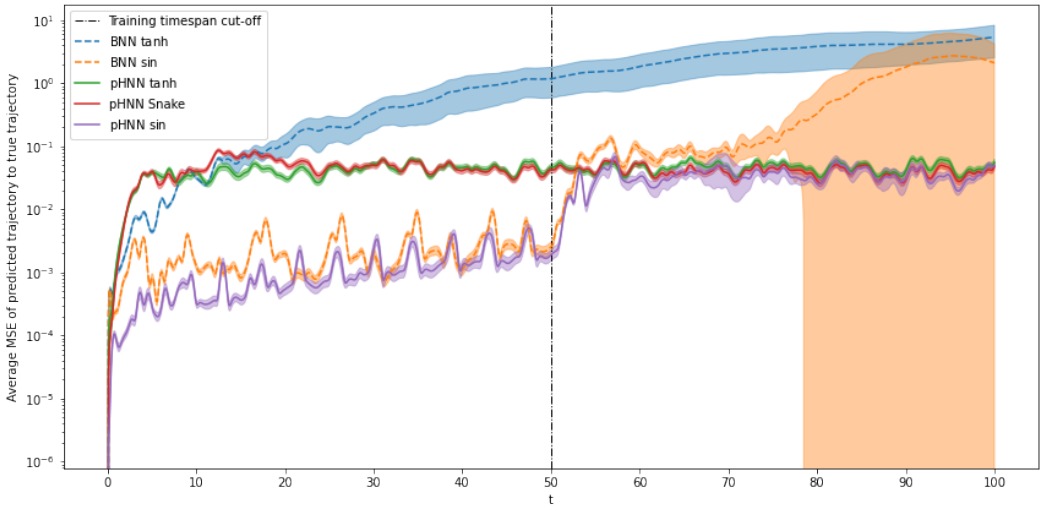

Figure 10. Mean of MSE of predictions for different modelling approaches with 2 SE (1 SE for BNN-tanh) confidence intervals against time.

7 Discussion

7.1 Limitations of our experimental settings

For all the modelling approaches we did not fine-tune model sizes i.e the quantity of layers and neurons within the layers, which could impact overfitting and other performance metrics of the models.
The performance of the pHNN model is sensitive to changes in regularisation parameters [9], as usual in penalised optimisation. The performed fine-tuning cannot grant an optimal choice.

7.2 Summary of findings

We have tested two physics-guided neural network approaches for modelling systems of coupled Duffing oscillators. The regular HNN approach proved to be viable for modelling unforced and undamped coupled Duffing oscillators, yielding accurate predictions for the time-evolving systems for testing and validation initial conditions in most of the cases we considered. For all systems, besides the system with a single oscillator, the HNN approach showed a statistically significant improvement over the baseline, non-Hamiltonain approach, resulting in more accurate models more consistently. Our deeper analysis of the 1-dimensional system model showed that the HNN model was accurate only locally, i.e in the regions around the data points from the training dataset.

We also performed experiments with a neural network architecture based on port-

Hamiltonian formalism, focused on a system of two coupled, forced and damped oscillators. Two modifications to the model architecture were proposed, namely two alternative activation functions in the first layer of the time-dependent part of the model. We tested the performance of the pHNN models over a much longer timespan, compared to the article that introduced the technique. In our results, all tested pHNN approaches failed to extrapolate to the time values not present in the training dataset. For the timespan observed in the training dataset, pHNN approach with $\tanh$ activation function in all layers also resulted in poor models, and the proposed modification with the Snake activation function did not provide an improvement in accuracy. The modification of pHNN and the baseline approach with sin activation function resulted in a great improvement in accuracy for the timespan present in training data, with both approaches yielding viable predictions, and generalising to the unseen initial conditions well. From the two, the modified pHNN-$\sin$ approach showed a statistically significant improvement in accuracy over the BNN-$\sin$ approach for most of the timespan, showing the best performance out of all tested approaches.

7.3 Recommendations for future research

The experiments on the unforced systems left some questions unanswered. For example, what are the general effects of the dimensionality of the modelled system on the accuracy, and generalisation performance of the model? Answering this question could yield a practical limit to the dimensionality of system for which HNN-based approaches are viable. Another research avenue would be determining how the accuracy of the model for a system with given dimensionality scales with the size of hidden layers, or with the amount or density of training data. From this, recommendations could be made for the optimal sizes of network layers and datasets for systems of given dimensionalities. In our research we used points along trajectories in the phase space as the training dataset, alternative ways of selecting training data could be explored, perhaps yielding better generalisation.

For the pHNN approach, we highlighted its shortcomings when presented with data on a large timespan. Here, regularisation methods for the proposed modification with sin activation function could be explored in order to improve generalisation performance.

The above-mentioned open research issues could be addressed by a formal analysis of the considered approaches, whose complexity make it out of the scope of this thesis.

References

[1] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

[2] H. Fan, L. Wang, Y. Du, Y. Wang, J. Xiao, and X. Wang, "Learning the dynamics of coupled oscillators from transients," *Physical Review Research*, vol. 4, no. 1, p. 013137, Feb. 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.4.013137

[3] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/089360809190009T

[4] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, "Hamiltonian neural networks for solving equations of motion," *Phys. Rev. E*, vol. 105, p. 065305, Jun 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.105.065305

[5] R. Wang, "Physics-guided deep learning for dynamical systems: A survey," *CoRR*, vol. abs/2107.01272, 2021. [Online]. Available: https://arxiv.org/abs/2107.01272

[6] A. Valenti, G. Jin, J. Léonard, S. D. Huber, and E. Greplova, "Scalable Hamiltonian learning for large-scale out-of-equilibrium quantum dynamics," *Physical Review A*, vol. 105, no. 2, p. 023302, Feb. 2022, arXiv: 2103.01240. [Online]. Available: http://arxiv.org/abs/2103.01240

[7] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, no. 1, p. 13890, Apr. 2017. [Online]. Available: http://www.nature.com/articles/ncomms13890

[8] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1906.01563

[9] 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, p. 034312, Sep 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.104.034312

[10] Y. M. Qing, Y. Ren, D. Lei, H. F. Ma, and T. J. Cui, "Strong coupling in two-dimensional materials-based nanostructures: a review," *Journal of Optics*, vol. 24, no. 2, p. 024009, jan 2022. [Online]. Available: https://doi.org/10.1088/2040-8986/ac47b3

[11] J. Awrejcewicz, *Bifurcation and Chaos in Coupled Oscillators*. WORLD SCIENTIFIC, Mar. 1991. [Online]. Available: http://www.worldscientific.com/worldscibooks/10.1142/1342

[12] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, "Forecasting hamiltonian dynamics without canonical coordinates," 2020.

[13] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics (3rd ed.).*

[14] B. Kröse and P. van der Smagt, *An Introduction to Neural Networks*. University of Amsterdam, 1996. [Online]. Available: https://www.researchgate.net/publication/272832321_An_introduction_to_neural_networks

[15] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009, oCLC: ocn237325326.

[16] S. Tõkke, "Kosmoloogiliste mudelite uurimine dünaamiliste süsteemide meetoditega," *Tartu Ülikool*, 2018.

[17] N. Atamanova, "Galaktikate filamendid kui dünaamilised struktuurid: teooria võrdlus vaatlustega," *Tartu Ülikool*, 2021.

[18] G. Leede, "Modifitseeritud teleparalleelse gravitatsiooniteooria uurimine dünaamiliste süsteemide meetodil," 2016.

[19] H. L. Smith, "Monotone dynamical systems: Reflections on new advances applications," *DISCRETE AND CONTINUOUS DYNAMICAL SYSTEMS, 10.3934/dcds.2017020*, 2017.

[20] G. Deco, M. L. Kringelbach, V. K. Jirsa, and P. Ritter, "The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core," *Scientific Reports*, vol. 7, p. 3095, Jun. 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5465179/

[21] R. T. Schroeder, J. L. Croft, and J. E. A. Bertram, "Evaluating the energetics of entrainment in a human–machine coupled oscillator system," *Scientific Reports*, vol. 11, no. 1, p. 15804, Aug. 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-95047-x

[22] S. P. Parsons and J. D. Huizinga, "Modulation of contractions in the small intestine indicate desynchronization via supercritical Andronov–Hopf bifurcation," *Scientific Reports*, vol. 10, no. 1, p. 15099, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-71999-4

[23] Y. Kuramoto, "Chemical oscillations, waves, and turbulence," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 1986. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-69689-3

[24] Y. Ikeda, H. Aoyama, Y. Fujiwara, H. Iyetomi, K. Ogimoto, W. Souma, and H. Yoshikawa, "Coupled Oscillator Model of the Business Cycle with Fluctuating Goods Markets," *Progress of Theoretical Physics Supplement*, vol. 194, pp. 111–121, 05 2012. [Online]. Available: https://doi.org/10.1143/PTPS.194.111

[25] S. Ceron, M. A. Kimmel, A. Nilles, and K. Petersen, "Soft robotic oscillators with strain-based coordination," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7557–7563, 2021.

[26] M. Brešar, P. Boškoski, and M. Horvat, "Detection of coupling in Duffing oscillator systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 6, p. 063130, Jun. 2021. [Online]. Available: https://aip.scitation.org/doi/10.1063/5.0050790

[27] S. Rajasekar and S. P. Raj, "The Painlev property, integrability and chaotic behaviour of a two-coupled Duffing oscillators," p. 16.

[28] C. Liu and L. Dong, "Physics-based control education: energy, dissipation and structure assignments," *European Journal of Physics*, vol. 40, 02 2019.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[31] L. Ziyin, T. Hartwig, and M. Ueda, "Neural Networks Fail to Learn Periodic Functions and How to Fix It," Oct. 2020, number: arXiv:2006.08195 arXiv:2006.08195 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2006.08195

A Radial distributions of datapoints for the unforced and undamped systems

The radial coordinate for each point is calculated as:

$$r = \sqrt{\sum_{i=1}^{2N} \eta_i^2}.$$

The distributions of data points for chosen datasets, as shown in Fig. A.1, help visualise high-dimensional data used in our experiments. The choice of the radial coordinate as reference point was motivated by the almost circular shape of orbits, which are often observed in oscillators.

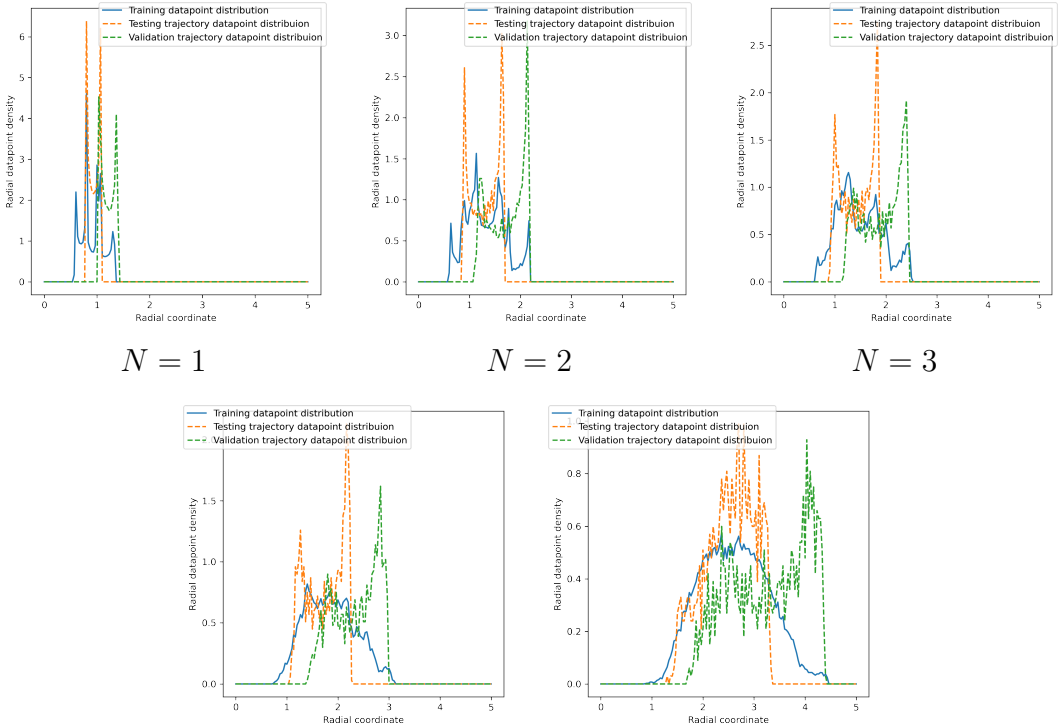

Figure A.1. Radial distributions of data points for considered systems

B Graphs of 1-st oscillator coordinate predicted with HNN for the unforced and undamped systems

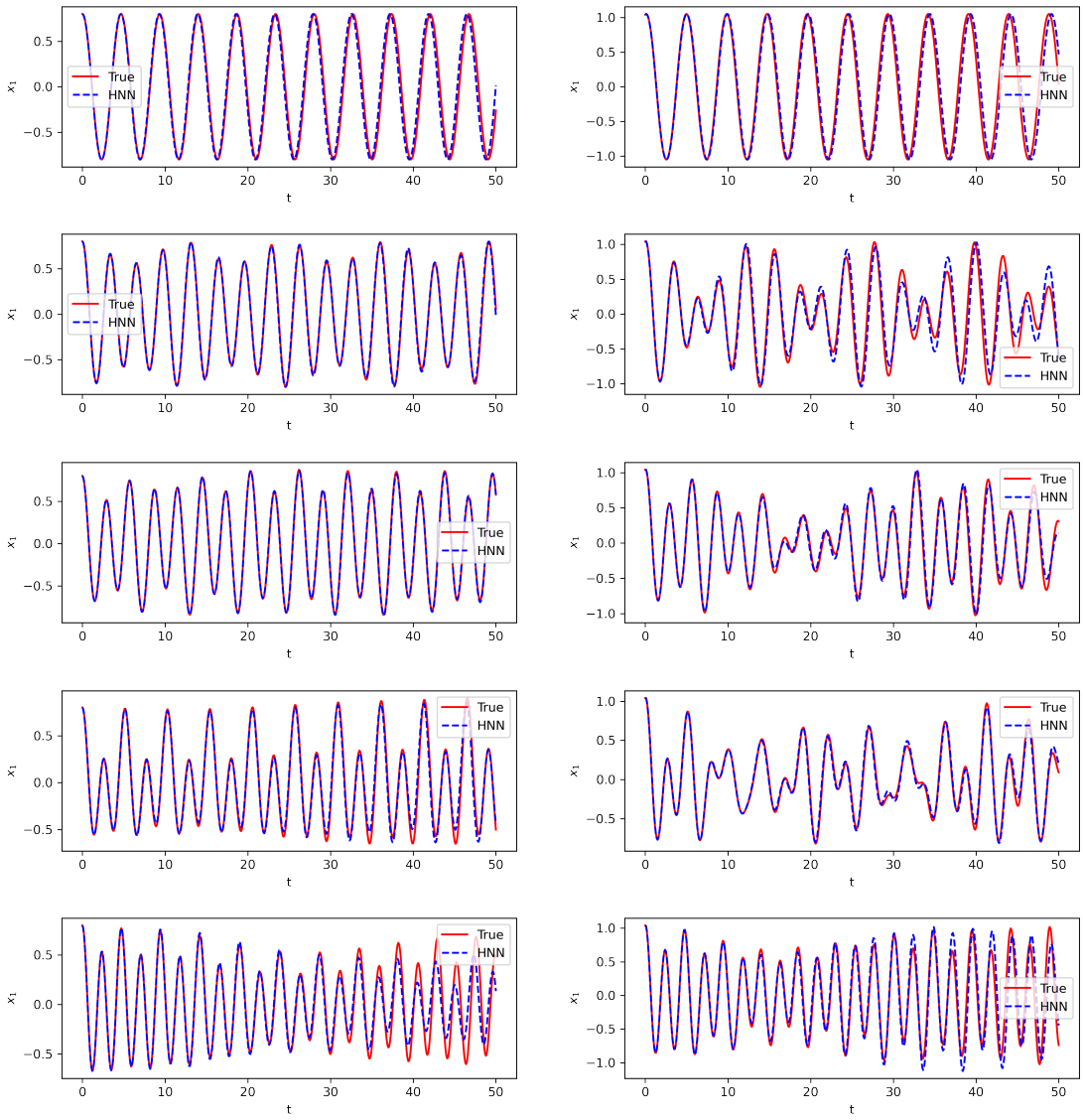

Figure B.1. Predicted (HNN) and True x_1 vs time for the test (left) and validation (right) initial conditions, ordered from top to bottom from system $N = 1$ to system $N = 5$.

C Graphs of 1-st oscillator velocity predicted with HNN for the unforced and undamped systems

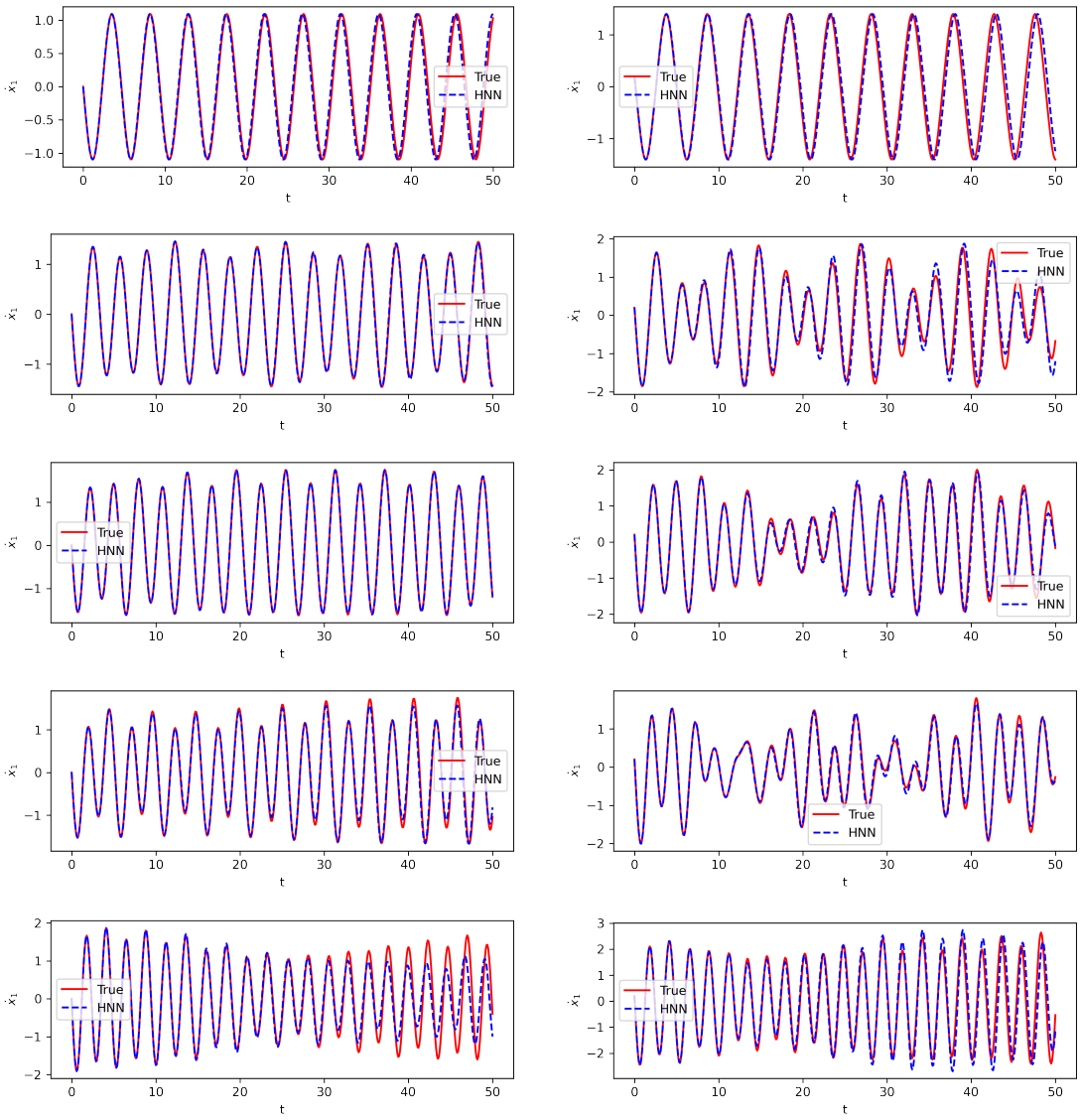

Figure C.1. Predicted (HNN) and True $\dot{x}_1$ vs time for the test (left) and validation (right) initial conditions, ordered from top to bottom from system $N = 1$ to system $N = 5$.

D Additional plots for coordinates and velocities of the first oscillator with different approaches for the forced and damped system

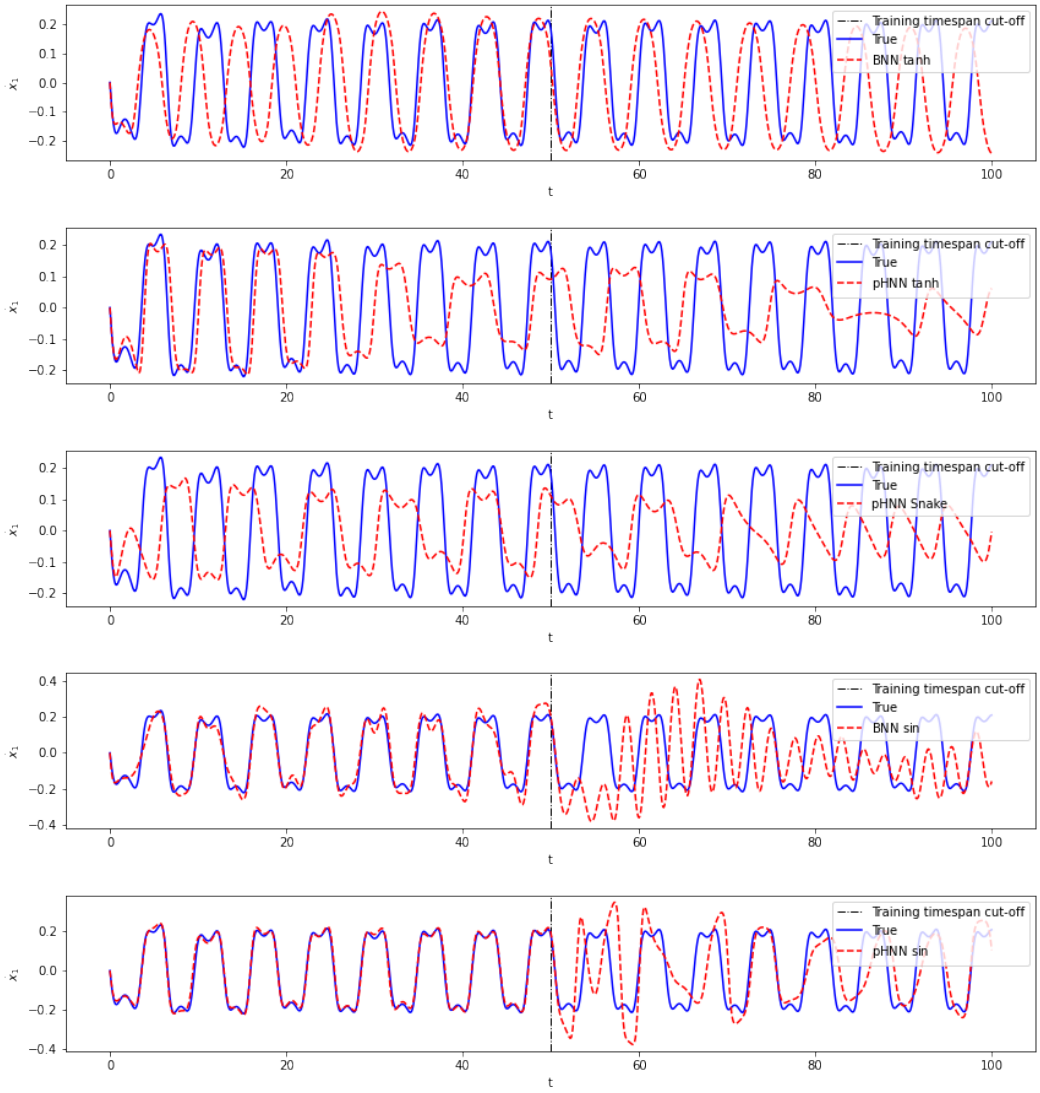

Figure D.1. Predicted and True $\dot{x}_1$ vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-tanh, HNN-tanh, HNN-Snake, BNN-sin, HNN-sin.

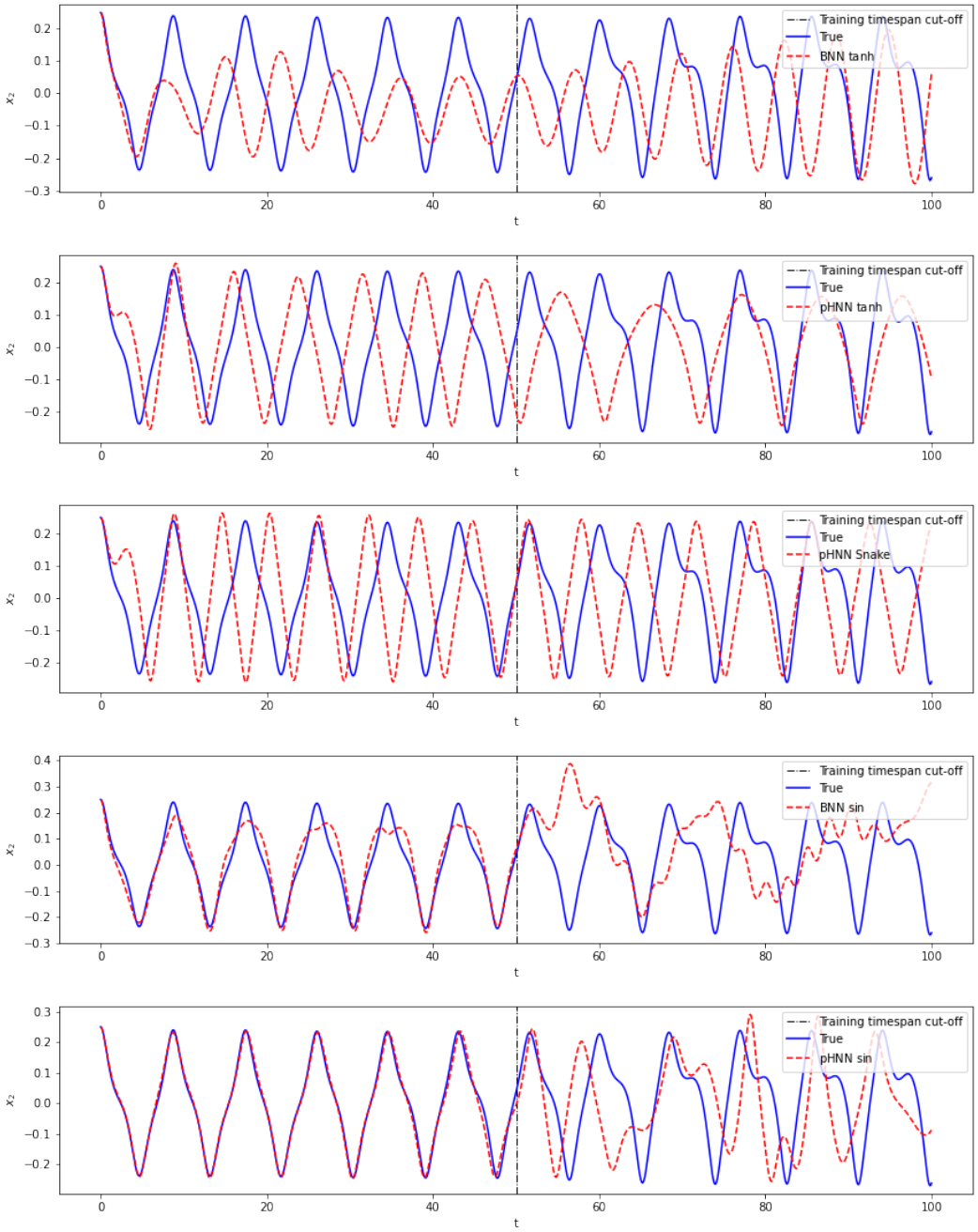

Figure D.2. Predicted and True x_2 vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-tanh, HNN-tanh, HNN-Snake, BNN-sin, HNN-sin.

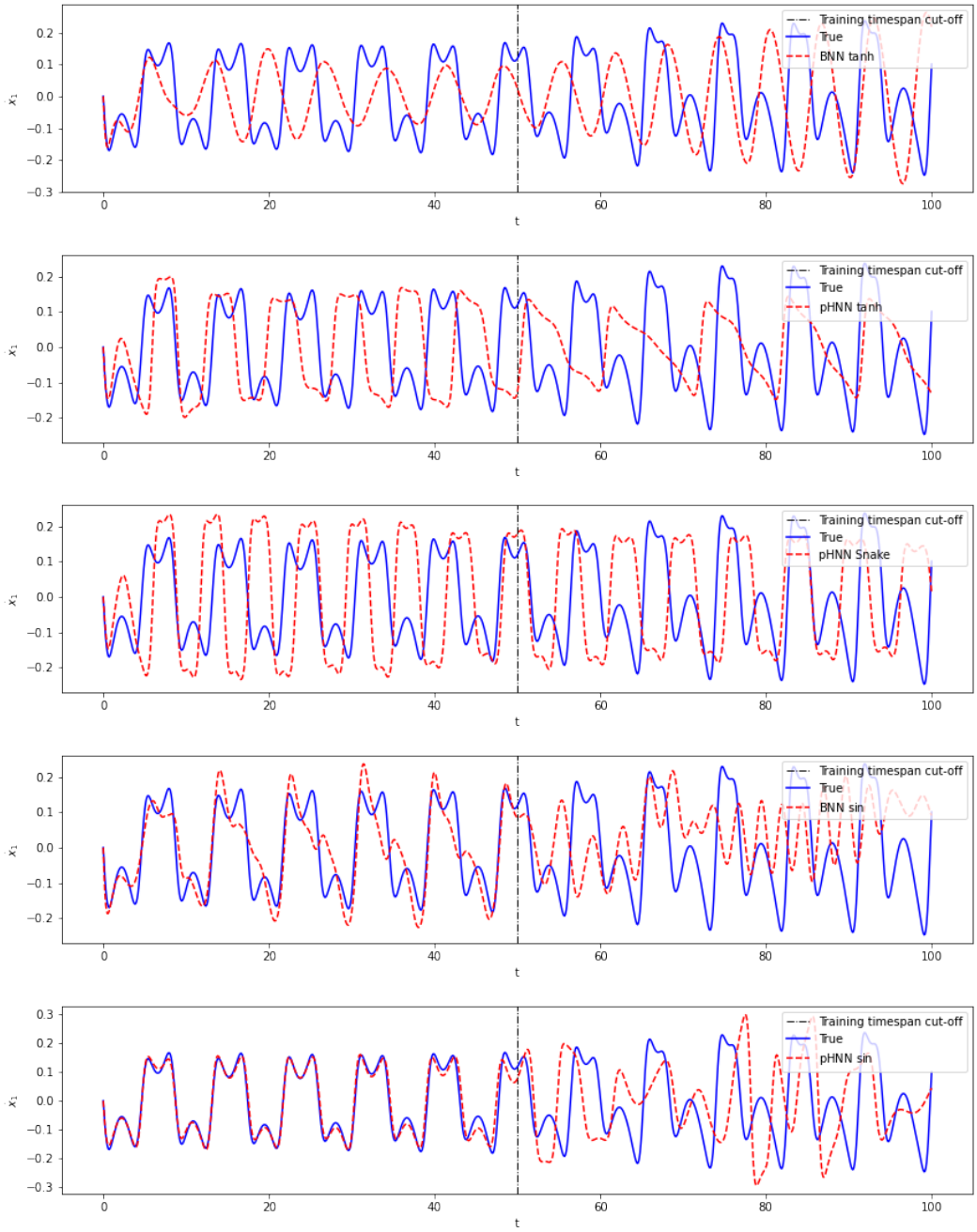

Figure D.3. Predicted and True $\dot{x}_2$ vs time for initial conditions $\eta_{0\text{test}}$ for BNN and pHNN approaches with varied activation function. Ordered from top to bottom: BNN-tanh, HNN-tanh, HNN-Snake, BNN-sin, HNN-sin.

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Gordei Pribõtkin**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Hamiltonian Neural Networks for coupled Duffing oscillators,

   supervised by Dr. Stefania Tomasiello.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons license CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work, from August 30, 2025 until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive license does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gordei Pribõtkin
29.05.2022