

Research paper

Regularized Action-conditioned Hamiltonian Generative Networks for Dynamics Learning with External Control

Arne Troch ^{*}, Kevin Mets , Siegfried Mercelis 

University of Antwerp - Imec, IDLab - Faculty of Applied Engineering, Sint-Pietersvliet 7, Antwerp, 2000, Belgium

ARTICLE INFO

Keywords:

Physics-informed machine learning
Hamiltonian neural networks
Dynamics learning
Control systems

ABSTRACT

Recent advancements in physics-informed machine learning have successfully leveraged Hamiltonian mechanics to learn system dynamics from data. In this work, we build upon Action-Conditioned Hamiltonian Generative Networks (ACHGN), a framework that extends these principles to non-canonical inputs and systems subject to external control. We identify a set of architectural changes from recent literature which contribute to significantly improved dynamics predictions. This novel ACHGN++ architecture generally outperforms ablations and baselines on a set of seven dissipation-free classical control tasks with dynamics of varying complexity. In addition, we highlight a limitation of the architecture and propose physics-informed regularization to enforce separation between the learned dynamics and external forces. We demonstrate that this regularization not only improves long-term prediction but also ensures the learned dynamics align more strongly with the intended Hamiltonian structure, as quantified by two novel evaluation metrics introduced in this work. Finally, we show that ACHGN++ is capable of learning an interpretable latent phase space which behaves similarly to real canonical coordinates on a pendulum system.

1. Introduction

Accurately modeling the dynamics of complex systems is a fundamental challenge across diverse scientific and engineering disciplines, crucial for applications ranging from prediction to control. Traditionally, this task relies on first-principle models, which necessitate significant domain expertise to derive the governing differential equations that accurately describe the system. Recent advances in machine learning offer an alternative, enabling the use of neural networks to learn the underlying dynamics of a system solely from observational data [1–3]. These purely data-driven approaches can be used when little or no prior knowledge of the system is available, provided sufficient data. However, they are subject to the inherent limitations of conventional neural networks, including limited generalization capabilities, susceptibility to overfitting, and a heavy reliance on extensive datasets [4].

In response to these limitations, the promising field of *physics-informed machine learning* seeks to incorporate physical laws directly into the learning process [5,6]. Such approaches aim to reduce the risks associated with purely data-driven learning by integrating data with additional information obtained by enforcing physical laws. Multiple approaches have been proposed, most notably Physics-Informed Neural Networks (PINNs) [7], which encode governing differential equations into the loss function of the learning process, and specialized net-

work architectures, which inherently embed physical principles. This second category is particularly appealing due to its reduced reliance on domain knowledge. While broad applications of this concept have been common practice in the form of Convolutional Neural Networks (CNNs) [8] and Recurrent Neural Networks (RNNs) [9], recent trends have shifted towards more powerful physical priors. Literature has explored the integration of Lagrangian mechanics [10,11], Hamiltonian mechanics [12,13] and Koopman theory [14,15] into the learning process. These core ideas have been further extended to include contact dynamics [16], explicit constraints [17] and energy dissipation [18,19]. In particular, approaches such as Hamiltonian Neural Networks (HNNs) [12] have gained prominence for their ability to learn energy-conserving dynamics by enforcing the symplectic nature of Hamiltonian dynamics on the learning process. Building on the success of Variational Autoencoders (VAEs), Hamiltonian Generative Networks (HGNs) [13] extend the concept of HNNs to learn latent Hamiltonian dynamics from non-canonical, high-dimensional inputs, such as images. This approach significantly increases applicability by eliminating the need for the input data to be presented in canonical coordinates. Subsequently, this framework was extended to systems subject to external forces [20]. This extension introduced Action-Conditioned Hamiltonian Generative Networks (ACHGNs), which learn a mapping between control inputs and their impact on the latent Hamiltonian dynamics. This work builds on

^{*} Corresponding author.

E-mail addresses: arne.troch@uantwerpen.be (A. Troch), kevin.mets@uantwerpen.be (K. Mets), siegfried.mercelis@uantwerpen.be (S. Mercelis).

the ACHGN framework to further enhance its dynamics modeling capabilities and improve our understanding of the learned latent dynamics. Additionally, this work proposes a regularization approach aimed to address a key identifiability challenge in controlled Hamiltonian systems: *disambiguating energy-conserving dynamics from external forces*. Our contributions are summarized as follows:

- We propose a dataset¹ consisting of seven dissipation-free classical control tasks of varying complexity designed to evaluate the performance of dynamics learning in the presence of external control from low-dimensional, non-canonical inputs.
- We introduce ACHGN + +, improving the original architecture by systematically evaluating various modifications proposed in recent literature [13]. We demonstrate its dynamics prediction performance by comparing with relevant baselines and ablations on the proposed dataset.
- We introduce a novel regularization approach aimed to enforce a clear separation between the learned dynamics and the effects of the external control. We validate the effectiveness of this regularization using traditional reconstruction metrics and two newly proposed metrics designed to evaluate Hamiltonian behavior in learned dynamics models.
- We demonstrate that ACHGN + + models learn an interpretable latent space, with latent phase-space variables behaving similarly to canonical coordinates. We show that this interpretability is dependent on the complexity of the system, with systems subject to complex dynamics leading to less interpretable latent spaces.

2. Background

This section introduces the fundamental concepts necessary to understand the contributions of this work. We first define Hamiltonian systems, outlining their core properties and time evolution. Subsequently, we show how the dynamics of Hamiltonian systems can be learned from data using HNNs and how this approach can be extended to various input modalities via HGNs. Finally, we discuss the extension of Hamiltonian systems to include external control inputs and how this can be modelled using ACHGNs.

2.1. Hamiltonian systems

Hamiltonian systems are characterized by a set of canonical coordinates $\mathbf{q} \in \mathbb{R}^n$ and conjugate canonical momenta $\mathbf{p} \in \mathbb{R}^n$, which evolve over time within a phase space $\mathbf{s} = (\mathbf{q}, \mathbf{p})$. This time evolution represents the system's dynamics, governed by a Hamiltonian function $\mathcal{H}(\mathbf{q}, \mathbf{p}, t) : \mathbb{R}^{2n} \rightarrow \mathbb{R}$. The scalar output of this function typically, though not necessarily, represents the total energy of the system as a function of the state space and time. The time evolution of the system is described by Hamilton's equations of motion:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}. \quad (1)$$

Any set of phase space coordinates $(\mathbf{q}, \mathbf{p})$ satisfying Eq. (1) is considered canonical and can be used to describe the time evolution of the system. The symplectic structure of Eq. (1) ensures the preservation of phase space volume over time, effectively representing energy-conserving systems:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \frac{d\mathbf{p}}{dt} = 0. \quad (2)$$

When the energy in a system is separable, meaning the kinetic energy depends solely on the momentum $\mathbf{p}$ and the potential energy depends solely on the position $\mathbf{q}$, the Hamiltonian can be expressed as the sum of two independent functions:

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + T(\mathbf{p}). \quad (3)$$

2.2. Learning Hamiltonian dynamics from data

The time evolution of a system's dynamics is often described using a first-order Ordinary Differential Equation (ODE):

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^{2n}, \quad t \in \mathbb{R}. \quad (4)$$

The field of dynamics learning considers learning this time evolution g from data captured from the system, without access to the underlying dynamics. Typically, this problem is formulated as learning a parameterized function f_θ to approximate the true dynamics: $f_\theta \approx g$. Traditional approaches addressed this by learning to predict the subsequent state of the system given the current one [21,22]:

$$\mathbf{x}_{t+1} = f_\theta(\mathbf{x}_t), \quad (5)$$

or by learning to predict the residual between two consecutive states [23]:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + f_\theta(\mathbf{x}_t). \quad (6)$$

However, these discrete-time approaches neglect two critical aspects of physical dynamics: continuous time evolution and the preservation of invariant quantities. To address the first issue, techniques such as Neural ODEs [24] learn to predict the time derivative of the current state:

$$\frac{d\mathbf{x}}{dt} = f_\theta(\mathbf{x}_t). \quad (7)$$

This formulation allows the learned dynamics to be interpreted as a continuous-time system, which can be integrated using numerical ODE solvers [25,26]. The second issue can be addressed through physics-informed techniques, which impose known physical laws or structural constraints on the learned dynamics. HNNs [12] embed the energy conservation properties of Hamiltonian dynamics into the learning process by parameterizing the Hamiltonian using a neural network $\mathcal{H}_\theta$. Given an initial state $(\mathbf{q}_0, \mathbf{p}_0)$, the time evolution of a Hamiltonian system is fully defined by its Hamiltonian function via Eq. (1). Consequently, the dynamics learned by HNNs are described as:

$$\dot{\mathbf{q}}_{t+1} = \mathbf{q}_t + \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} dt, \quad \dot{\mathbf{p}}_{t+1} = \mathbf{p}_t - \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} dt. \quad (8)$$

The partial derivatives in Eq. (8) are computed using in-graph gradients of the Hamiltonian neural network with respect to its inputs. The original formulation of HNNs [12] proposed training the Hamiltonian network by minimizing the difference between the time derivatives predicted by the learned dynamics and the true time derivatives of the system:

$$\mathcal{L}_{HNN} = \left\| \frac{\partial \mathbf{q}}{\partial t} - \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} \right\| + \left\| \frac{\partial \mathbf{p}}{\partial t} + \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} \right\|. \quad (9)$$

However, since true time derivatives are often unavailable in practice, subsequent works have proposed alternative training objectives. These include using approximate derivatives obtained through finite differences [27,28]:

$$\mathcal{L}_{HNN} = \left\| \frac{\mathbf{q}_{t+1} - \mathbf{q}_t}{dt} - \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} \right\| + \left\| \frac{\mathbf{p}_{t+1} - \mathbf{p}_t}{dt} + \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} \right\|, \quad (10)$$

or minimizing the error between predicted and observed states over time [13,29]:

$$\mathcal{L}_{HNN} = \|\mathbf{q}_{t+1} - \hat{\mathbf{q}}_{t+1}\| + \|\mathbf{p}_{t+1} - \hat{\mathbf{p}}_{t+1}\|, \quad (11)$$

where $\hat{\mathbf{q}}_{t+1}, \hat{\mathbf{p}}_{t+1}$ are the predicted states obtained from Eq. (8). Results have shown that HNNs are able to effectively learn energy-conserving dynamics given a set of datapoints sampled from the original system, provided that these datapoints are in the form of a set of canonical coordinates: $\mathbf{x} = (\mathbf{q}, \mathbf{p})$. However, such canonical datapoints are often unavailable in real-world scenarios, where systems are typically observed through high-dimensional sensors (e.g., cameras) or measurements that do not correspond to canonical coordinates (e.g., actuator states).

To overcome this limitation, the concept of *latent Hamiltonian dynamics* was introduced. This approach extends the applicability of HNNs to

¹ Available at <https://huggingface.co/datasets/Atroch/ControlEnvs>

diverse, non-canonical, input modalities by learning a mapping from the observed data space to a latent phase space, where a learned Hamiltonian function effectively governs the dynamics. This concept was first introduced as HGNs [13], which use an autoencoder architecture to encode non-canonical inputs into a canonical latent phase space. The proposed architecture consists of four separate networks:

$$\begin{aligned} \mathbf{z}_t &\sim q_\theta(\cdot | \mathbf{x}_t) && \text{(encoder)} \\ \tilde{\mathbf{s}}_t &= f_\theta(\mathbf{z}_t) && \text{(transformer)} \\ \tilde{\mathbf{s}}_{t+1} &= \text{Integrator}(H_\theta(\tilde{\mathbf{s}}_t)) && \text{(dynamics)} \\ \hat{\mathbf{x}}_t &= d_\theta(\tilde{\mathbf{s}}_t) && \text{(decoder),} \end{aligned} \quad (12)$$

where q_θ encodes the observed data into a latent representation, f_θ transforms this representation into a latent phase space, H_θ models the Hamiltonian dynamics within this space, and d_θ decodes the latent states back to the original data space. The latent phase space is constructed by arbitrarily partitioning the transformed latent representation $\tilde{\mathbf{s}}_t$ into generalized coordinates $\tilde{\mathbf{q}}_t$ and conjugate momenta $\tilde{\mathbf{p}}_t$. The overall optimization objective of this architecture is defined as:

$$\mathcal{L}_{HGN} = \sum_{t=0}^T [\log d_\theta(\hat{\mathbf{x}}_t | \tilde{\mathbf{s}}_t)] - KL[\mathbf{z}_t || p(\mathbf{z}_t)], \quad (13)$$

where T denotes the length of the training sequence. The log-term represents the reconstruction log-likelihood, while the KL-term regularizes the latent encoding to approximate a prior distribution $p(\mathbf{z})$ (typically a unit normal distribution). This framework integrates the principles of HNNs with the generative capabilities of VAEs [30], enabling the learning of latent Hamiltonian dynamics from non-canonical observations.

2.3. Action-conditioned Hamiltonian Generative Networks

When a system is subject to external control inputs, typically generated by actuators, the assumption of energy conservation is violated, rendering the dynamics described in Eq. (1) insufficient. To address this, we can formulate an extended version of Hamiltonian dynamics that incorporates these external forces:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}} + g(\mathbf{q})\mathbf{u}, \quad (14)$$

where $\mathbf{u} \in \mathbb{R}^m$ represents the external control inputs and $g(\mathbf{q}) \in \mathbb{R}^{n \times m}$ is an input matrix defining how these controls influence the generalized momenta. We follow previous works in assuming external control is affine and only influences changes in generalized momenta [29]. When $\mathbf{u} = \mathbf{0}$, the dynamics reduce to the classical Hamiltonian form, preserving the original energy conservation properties. Conversely, when $\mathbf{u} \neq \mathbf{0}$, the input matrix $g(\mathbf{q})$ ² determines the impact of each control element on the generalized momenta. The formulation in Eq. (14) describes a system where all changes in energy are caused by the external control. Further extensions of this formulation, such as those incorporating energy dissipation leading to *pseudo-Hamiltonian* or *port-Hamiltonian* systems [18,19], exist but are beyond the scope of this work, which focuses on environments without energy dissipation. In previous work, the authors have introduced an extension to the HGN architecture to account for external control inputs, referred to as ACHGN [20]. The dynamics update step in Eq. (12) is modified to include an embedding of the external control $\mathbf{u}_t$:

$$\tilde{\mathbf{s}}_{t+1} = \text{Integrator}(H_\theta(\tilde{\mathbf{s}}_t) + A_\theta(\tilde{\mathbf{s}}_t, \mathbf{u}_t)), \quad (15)$$

where A_θ is a neural network that models the effect of the control input $\mathbf{u}_t$ on the latent phase space $\tilde{\mathbf{s}}_t$, effectively representing the input matrix $g(\mathbf{q})$ within the latent dynamics. Consequently, the latent phase space evolves according to:

$$\frac{d\tilde{\mathbf{q}}}{dt} = \frac{\partial H_\theta}{\partial \tilde{\mathbf{p}}}, \quad \frac{d\tilde{\mathbf{p}}}{dt} = -\frac{\partial H_\theta}{\partial \tilde{\mathbf{q}}} + A_\theta(\tilde{\mathbf{s}}, \mathbf{u}), \quad (16)$$

which represents latent Hamiltonian dynamics subject to external control. The optimization objective is the same as in Eq. (13), with the dynamics now defined by Eq. (15).

3. Experimental setup

This section outlines the experimental framework employed throughout this work. First, we describe the datasets used to train the various dynamics models, detailing the systems considered and the data collection process. Second, we define the evaluation metrics used to assess model performance. Third, we detail the training specifications for the ACHGN architecture, evaluating the base architecture proposed in [20] and comparing its performance against various modifications proposed in recent literature [31].

3.1. Datasets

All datasets in this work are collected from the control environments introduced by [32]. These environments utilize first-principle models to simulate seven classical control tasks of varying complexity, ranging from a fully-actuated ($g(\mathbf{q})$ is full column rank) pendulum swing-up to a complex underactuated ($g(\mathbf{q})$ is rank-deficient) Cart3Pole balancing task. Since the observed environment states are not provided in the form of canonical coordinates, they are not suitable for the direct application of HNNs and require the autoencoder architecture of HGNs. The environments are subject to control inputs $\mathbf{u} \in [-1, 1]$, which are scaled to apply a force $F = g(\mathbf{q})\mathbf{u}$ to the system velocities. The input matrix $g(\mathbf{q})$ is unknown, control-affine and may contain zero elements.

Data is collected by simulating the system dynamics with random control inputs starting from randomly initialized states. The range of initial conditions was selected to ensure adequate state space coverage when unrolling forward.³ The specific initial conditions for each environment are detailed in Appendix A. This broad initialization is critical for systems such as the pendulum, which are prone to quickly stabilize around equilibrium points. Training data is generated by simulating trajectories for 300 steps. From each unrolled trajectory, 10 training sequences of 64 steps are sampled until a total of 50,000 training sequences is collected. Similarly, 20,000 test sequences of length 125 are sampled from longer trajectories of 1000 steps, with 5 test sequences sampled per trajectory. Additionally, 10 trajectories of length 1000 are reserved for evaluating long-term dynamics predictions. Finally, 5000 test sequences with zero-control inputs (i.e., $\mathbf{u} = \mathbf{0}$) are collected following the same sampling strategy as the test data. These sequences facilitate further analysis of the learned dynamics, specifically targeted at Hamiltonian behavior under energy-conserving conditions. A comprehensive overview of the datasets used in this work is presented in Table A.1.

3.2. Evaluation metrics

Evaluating learned dynamics typically relies on reconstruction error metrics, which calculate the difference between predicted and ground truth states. However, it has been demonstrated that such metrics do not provide sufficient insight into the quality of the learned dynamics [31,33]. In this section, we detail two standard metrics used in the literature to evaluate learned dynamics models. These metrics will be used to assess the performance of all learned dynamics in this work. In Section 5.2, we introduce two novel metrics specifically designed to evaluate Hamiltonian behavior in learned dynamics models.

MSE. We employ the MSE between predicted and ground truth states on an unseen test set as a primary measure of model performance, a

² $g(\mathbf{q})$ may be rank-deficient, leading to underactuated systems.

³ Example videos from each environment are displayed at <https://sites.google.com/view/achgn>.

Table 1

Reconstruction (**Rec**) and extrapolation (**Ext**) refer to the normalized MSE measured across the first 60 and subsequent 60 timesteps on the test set, respectively. VPT refers to the number of predicted timesteps before the normalized MSE exceeds a threshold λ of 0.9. MSE values are reported as multiples of 10^{-2} .

Environment	ACHGN			β -VAE			Leap Frog			No Transform			Separable			Learned Δt		
	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT
Pendulum	0.07	10.14	133	0.71	13.48	102	0.04	8.48	128	0.07	6.92	136	0.11	9.86	130	0.06	4.32	140
Reacher	0.12	0.78	210	0.82	4.86	194	0.15	0.78	226	0.15	0.68	242	0.10	1.82	231	0.10	0.76	218
CartPole	0.03	0.68	160	0.22	2.48	149	0.09	0.69	191	0.01	0.49	181	0.03	1.29	180	0.06	1.16	163
Acrobot	3.05	65.64	95	2.75	62.77	54	2.65	68.53	65	2.66	72.59	75	2.53	61.25	93	2.85	50.94	106
Cart2Pole	2.07	35.17	87	2.43	36.15	80	2.65	34.28	98	1.45	33.89	96	1.77	33.45	98	2.86	24.01	94
Cart3Pole	24.89	125.34	4	16.97	105.19	10	26.22	124.00	0	19.30	108.45	8	63.11	122.36	0	24.72	99.66	0
Acro3bot	81.75	143.53	0	70.88	124.89	0	77.17	135.29	0	24.54	129.75	7	62.89	125.31	0	24.07	111.95	0
Average	16.00	54.47	98	13.54	49.97	84	15.57	53.15	101	6.88	50.40	107	18.65	50.76	105	7.82	41.83	103

common practice in dynamics learning [13,29,33]. This metric quantifies how well the model predicts system states over a training sequence of length T , referred to as “reconstruction” performance. Following [31], we also calculate an “extrapolation” performance by unrolling the model for an additional T timesteps beyond the training trajectory length. To facilitate comparison across datasets, we normalize the MSE by the squared average ground truth state, as proposed by [34]. This leads to the following definitions:

$$\text{MSE}_{\text{recon}} = \frac{1}{T} \sum_{t=0}^T \frac{(\mathbf{x}_t - \hat{\mathbf{x}}_t)^2}{\mathbf{x}_t^2}, \quad (17)$$

$$\text{MSE}_{\text{extra}} = \frac{1}{T} \sum_{t=T}^{2T} \frac{(\mathbf{x}_t - \hat{\mathbf{x}}_t)^2}{\mathbf{x}_t^2},$$

where $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ denote the ground truth and reconstructed data at timestep t , respectively. The MSE scores presented in this work are averaged over the test set. While $\text{MSE}_{\text{recon}}$ indicates the model’s ability to fit the training data, $\text{MSE}_{\text{extra}}$ provides valuable insight in the ability of the model to perform beyond its training regime. Given that longer rollouts are susceptible to compounding errors, which significantly impact dynamics predictions [35], $\text{MSE}_{\text{extra}}$ is crucial for assessing the quality of the learned dynamics.

VPT. The VPT metric is motivated by the idea that models which effectively capture the underlying dynamics of the system should be able to predict over timespans significantly exceeding the training horizon [31,33,36]. VPT quantifies the duration for which model predictions remain within a specified error threshold λ of the ground truth. We unroll the learned dynamics along the long test trajectories and identify the timestep at which the prediction error exceeds λ :

$$\text{VPT} = \underset{t}{\text{argmin}} [\text{MSE}(\mathbf{x}_t, \hat{\mathbf{x}}_t) > \lambda]. \quad (18)$$

A higher VPT score indicates that the model maintains accurate dynamics predictions for a longer duration before significant divergence occurs. Note that we discard the first T timesteps of the results, since we aim to evaluate long-term prediction capabilities beyond the training regime. Consequently, the maximum achievable VPT score is 940.

3.3. Training specifications

All models are trained on sequences of $T = 60$ timesteps, using five stacked frames as input for inference. Since model predictions start from the reconstruction of the current state, $\hat{\mathbf{x}}_0 = d_\theta(q_\theta(\mathbf{x}_0))$, each training sequence consists of 64 total steps. Table E.16 lists the training and network hyperparameters which are kept constant across all experiments in this work. All results reported are averaged over three random seeds, standard deviations for each experiment are reported in Appendix E.

We conduct a study of the modifications proposed in HGN++ [13] within the context of ACHGNs to determine the optimal training proce-

dure for all further experiments in this work. Note that we do not evaluate the forward/backward training scheme introduced in [37] since backward rollouts require knowledge of the force applied during the transition $\tilde{\mathbf{s}}_i \rightarrow \tilde{\mathbf{s}}_{i+1}$, which is conditioned on $\tilde{\mathbf{s}}_i$: $F_i = A_\theta(\tilde{\mathbf{s}}_i, \mathbf{u}_i)$. In a reverse time setting, $\tilde{\mathbf{s}}_i$ is the target state and thus unavailable for calculating the required force. We evaluate the following modifications from [13]:

- **Training objective:** The original HGN (and ACHGN) implementation optimizes the objective in Eq. (13) using the GECO approach [38], whereas HGN++ proposes to use the β -VAE objective [39].
- **Integrator:** Both HGN and HGN++ utilize the leap-frog integrator [40] for unrolling Hamiltonian dynamics. Given that the systems in this work violate the energy-conservation assumption present in previous works, we test whether the leap-frog integrator maintains its performance advantage over a simple Euler integration scheme.
- **Latent transform:** HGN++ suggests removing the latent transform f_θ , treating the encoder output directly as the latent state.
- **Separable Hamiltonian:** Findings from [13] suggest that modeling the Hamiltonian $\mathcal{H}_\theta$ with separate networks for potential and kinetic energy (as in Eq. (3)) improves performance compared to a single network.

Additionally, we evaluate the impact of learning the integration timestep as a model parameter, as opposed to using a fixed value. This concept, originally proposed in [20] with preliminary positive results, is further validated here. In experiments with a fixed timestep, the original environment timestep, specified in Appendix A, is used.

The MSE scores presented in Table 1 demonstrate that most of the proposed modifications improve both reconstruction and extrapolation performance. Specifically, removing the latent transform and learning the timestep yield significantly better predictions. The VPT scores confirm this trend, showing an overall improvement in long-term prediction performance, with the most significant improvements again attributed to removal of the latent transform. Notably, both the leap-frog integrator and the separable Hamiltonian network also lead to better long-term predictions, despite showing similar or slightly worse MSE scores. We believe the divergence between short-horizon accuracy and long-term predictions can be attributed to the classical problem of overfitting to the training horizon, with modifications such as the leap-frog integrator and separable Hamiltonian constraining the model to limit this problem and better learn the underlying dynamics. Finally, we observe a significant discrepancy between the MSE and VPT results for the β -VAE objective, with MSE scores indicating improved reconstruction, while VPT results reveal faster divergence over longer trajectories. Based on these results, we incorporate all proposed modifications into the final architecture, referred to as ACHGN++. Given the inconclusive results of the two training objectives, we evaluate both GECO (γ -ACHGN++) and β -VAE (β -ACHGN++) approaches in combination with the other modifications in Section 4.

Table 2

Reconstruction (**Rec**) and extrapolation (**Ext**) refer to the normalized MSE measured across the first 60 and subsequent 60 timesteps on the test set, respectively. VPT refers to the number of predicted timesteps before the normalized MSE exceeds a threshold λ of 0.9. MSE values are reported as multiples of 10^{-2} .

Environment	ACHGN			β -ACHGN++			γ -ACHGN++			LatNODE			RGN			HGN		
	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT	Rec	Ext	VPT
Pendulum	0.07	10.14	133	0.16	2.67	404	0.04	4.16	113	0.29	2.41	369	0.29	3.20	382	1.19	19.24	107
Reacher	0.12	0.78	210	0.92	3.43	282	0.72	0.48	206	0.94	3.65	256	0.84	3.71	312	6.25	26.84	54
CartPole	0.03	0.68	160	0.14	0.73	710	0.12	0.18	224	0.20	0.67	517	0.19	0.89	409	2.75	13.42	58
Acrobot	3.05	65.64	95	2.03	27.07	100	2.65	48.19	85	2.96	23.97	89	4.00	45.51	45	35.01	102.98	9
Cart2Pole	2.07	35.17	87	0.97	13.94	100	1.78	21.95	103	1.24	13.40	104	1.18	13.52	81	3.77	42.56	52
Cart3Pole	24.89	125.34	4	17.39	93.59	2	16.65	98.09	14	15.54	86.31	8	25.10	90.91	0	31.47	114.07	0
Acro3bot	81.75	143.53	0	25.61	113.34	4	26.39	110.54	9	28.12	109.83	7	27.00	109.84	0	57.58	123.30	0
Average	16.00	54.47	98	6.75	36.40	229	6.91	40.51	108	7.04	34.32	193	8.37	38.22	176	19.72	63.20	40

4. Comparing ACHGN with baselines

This section presents a comprehensive evaluation of the ACHGN++ architecture against established dynamics learning baselines, utilizing the datasets and evaluation metrics defined in Section 3. We aim to assess the effectiveness of ACHGN++ in capturing the underlying dynamics of controlled systems, particularly in comparison to both physics-informed and purely data-driven approaches. First, we describe the models used for comparison and ablation. Next, we present and discuss the evaluation scores for each model. Finally, we provide a qualitative comparison of the learned dynamics by visualizing example trajectories for both controlled and uncontrolled scenarios.

4.1. Models

We compare two variants of the proposed ACHGN++ architecture with four baseline models. Each model employs the autoencoder architecture defined in Table E.16 for encoding and decoding observations, with only the latent dynamics model varying between approaches. These baselines serve as an ablation study for the various components of the ACHGN++ architecture, isolating the effects of the Hamiltonian structure, the action conditioning, and the modifications introduced in Section 3.

HGN. This architecture serves as an ablation for the action-conditioning component of ACHGN++. We achieve this by removing the force network A_θ from the architecture, resulting in dynamics that follow the energy-conserving Hamiltonian form described in Eq. (12). To ensure a fair comparison, we implement the same architectural modifications as in ACHGN++, creating a baseline similar to HGN++ [13], except for the forward-backward training scheme.

LatNODE. This model serves as an ablation for the Hamiltonian structure of ACHGN++. It represents a primarily data-driven approach, with the only “physics-informed” component being the continuous-time formulation of the dynamics. This approach has been previously shown to outperform the HGN architecture in energy-conserving settings [41] and therefore constitutes a strong benchmark for validating the dynamics learning capabilities of ACHGN++. LatNODE replaces the Hamiltonian network $\mathcal{H}_\theta$ and the force network A_θ with a single neural network f_θ that predicts the latent time derivative $d\tilde{s}/dt = f_\theta(\tilde{s}, \mathbf{u})$. The dynamics are unrolled using an explicit Runge-Kutta integrator of order 2, following the benchmark established in [41].

RGN. As a further ablation of the physics-informed components of ACHGN++, we implement a discrete-time variant of LatNODE. This results in a purely data-driven dynamics model, effectively establishing a performance benchmark for recurrent models without any physics-informed structure. In this model, the dynamics are represented by a discrete step function f_θ , predicting the change in the latent state $\Delta\tilde{s}_t = f_\theta(\tilde{s}_t, \mathbf{u}_t)$. We employ a residual formulation (as Eq. (6)) where

the next state is predicted as $\tilde{s}_{t+1} = \tilde{s}_t + f_\theta(\tilde{s}_t, \mathbf{u}_t)$, as this has been shown to improve performance over predicting the next state directly [41].

4.2. Results

Table 2 presents the quantitative results for all proposed models, with a computational cost comparison provided in Table E.17. First, we observe that both ACHGN++ variants outperform the baseline ACHGN architecture, demonstrating the effectiveness of the proposed architectural modifications. The results also indicate a strong advantage of the β -VAE training objective over the GECO approach. Most notably, the β -VAE objective increases the average VPT score of ACHGN by 133.7%, compared to an improvement of only 10.2% with the GECO objective. When comparing β -ACHGN++ to the ablation architectures, we observe that only LatNODE approaches similar performance levels, even achieving a lower average extrapolation MSE. However, LatNODE falls short in long-term prediction performance; while it marginally outperforms β -ACHGN++ in the three most complex environments (Cart2Pole, Cart3Pole, Acro3bot), it significantly underperforms in the simpler tasks. Notably, all proposed methods achieve low VPT and high MSE scores for the more complex tasks (Cart3Pole, Acro3bot), indicating that these systems are beyond the capabilities of the methods to effectively model. The discrete-time RGN baseline performs better than the unmodified ACHGN but is outperformed by both β -ACHGN++ and LatNODE across all metrics. Finally, the HGN baseline exhibits the weakest performance overall, highlighting the significant impact of external forces on the system dynamics.

Fig. 1 visualizes two example trajectories of 1000 timesteps on the pendulum system: one with external control inputs and one without. For each trajectory, we compare the ground truth states with the predictions from β -ACHGN++ and LatNODE, visualizing the angle, angular velocity, and phase space trajectory over time. In the controlled trajectory, β -ACHGN++ closely follows the ground truth in both amplitude and oscillation frequency, correctly predicting an increase in energy over time (evident in the increasing peaks). The phase-space trajectory further confirms this, as the trajectory predicted by β -ACHGN++ closely matches the phase-space volume of the ground truth. In contrast, LatNODE significantly underestimates the energy increase, resulting in lower amplitude oscillations and a smaller phase-space volume. This indicates an inability to fully capture the impact of control inputs on the system dynamics. In the zero-control trajectory, both models suffer from energy loss over time, deviating from the expected closed phase-space orbits of a Hamiltonian system. However, β -ACHGN++ exhibits significantly lower energy drift, showing only minimal deviations from the baseline trajectory, whereas LatNODE quickly spirals towards the origin, indicating a rapid loss of energy. While these results represent a single example trajectory, they provide qualitative insight into the differences in learned dynamics between the two models, highlighting the strengths of β -ACHGN++ in capturing both controlled and energy-conserving dynamics. More example trajectories are provided in Fig. D.1 in the Appendix.

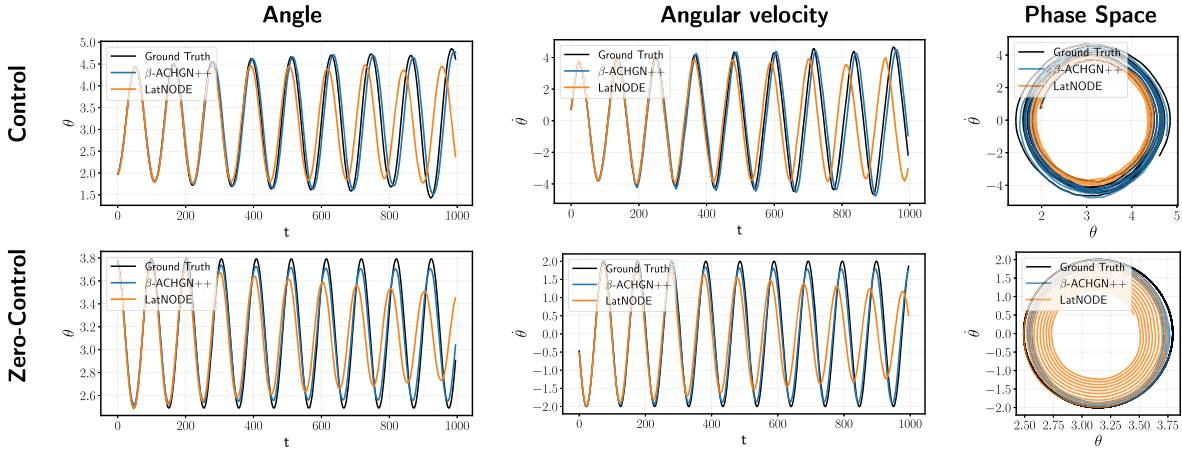


Fig. 1. Visualization of ground truth and predicted trajectories for both control and zero-control trajectories on the Pendulum system. The first two columns show the angle θ and angular velocity $\dot{\theta}$ dimensions over time, while the third column displays the phase space (θ vs. $\dot{\theta}$). Ground truth trajectories are shown in black, β -ACHGN++ predictions in blue and LatNODE predictions in orange. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Regularizing ACHGN

While ACHGN provides a structured framework for learning controlled Hamiltonian dynamics, the Hamiltonian network $\mathcal{H}_\theta$ and force network A_θ are not explicitly pushed towards their intended physical roles during training. Consequently, the model may learn a Hamiltonian that does not accurately represent the system's true conservative energy, relying instead on the force network to compensate for energy discrepancies. In this section, we introduce a regularization loss to the ACHGN training process, designed to enforce a clearer separation between the conservative and non-conservative components of the dynamics. Subsequently, we present two novel metrics aimed at quantifying the Hamiltonian nature of the learned dynamics. We then evaluate the impact of the proposed regularization on the dynamics learned by ACHGN++, analyzing both standard reconstruction metrics and the newly introduced Hamiltonian behavior metrics. Finally, we demonstrate the capacity of ACHGN++ models to learn interpretable latent spaces that closely resemble canonical coordinates, providing a strong advantage over purely data-driven models.

5.1. Regularizing the energy-work balance

When a system is subject to external control inputs, total energy is not conserved and the time derivative of the Hamiltonian described in Eq. (2) is no longer zero. The correct time derivative is obtained by substituting the dynamics from Eq. (14) into Eq. (2), yielding the following relationship between energy change and external control:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \cdot \mathbf{g}(\mathbf{q})\mathbf{u}. \quad (19)$$

The derivation of this identity is provided in Appendix B. This equation states that the rate of change of the Hamiltonian (i.e., the change in system energy) equals the power input from external forces, represented by the dot product of the generalized momenta and the applied forces. If the learned Hamiltonian $\mathcal{H}_\theta$ and force network A_θ accurately capture their respective components of the system's dynamics, they should satisfy the analogous relation:

$$\frac{d\mathcal{H}_\theta}{dt} = \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} \cdot A_\theta(\tilde{\mathbf{s}}, \mathbf{u}). \quad (20)$$

For a discrete transition $t \rightarrow t+1$ with integration step Δt , we use the first-order approximation:

$$\mathcal{H}_\theta^{t+1} - \mathcal{H}_\theta^t \approx \Delta t \frac{\partial \mathcal{H}_\theta^t}{\partial \mathbf{p}_t} \cdot A_\theta(\tilde{\mathbf{s}}_t, \mathbf{u}_t). \quad (21)$$

We incorporate this discrete relation as a regularization term in the ACHGN++ training objective, defining the Energy-Work Balance (EWB) loss as:

$$\mathcal{L}_{\text{EWB}} = \frac{\alpha}{T} \sum_{t=0}^T \left| \mathcal{H}_\theta^{t+1} - \mathcal{H}_\theta^t - \Delta t \frac{\partial \mathcal{H}_\theta^t}{\partial \mathbf{p}_t} \cdot A_\theta(\tilde{\mathbf{s}}_t, \mathbf{u}_t) \right|, \quad (22)$$

where $\mathcal{H}_\theta^t = \mathcal{H}_\theta(\tilde{\mathbf{s}}_t)$, $\mathcal{H}_\theta^{t+1} = \mathcal{H}_\theta(\tilde{\mathbf{s}}_{t+1})$ and α is a hyperparameter controlling the impact of the regularization on the learning process. Here, Δt denotes the effective integration step used by the model at transition t : in fixed-timestep experiments, Δt equals the environment step size; in learned-timestep experiments, Δt equals the learned integration step at that point in the training process. This loss penalizes discrepancies between the change observed from the learned Hamiltonian and the change derived from the work done by external forces. By minimizing this loss during training, we encourage the model to learn a Hamiltonian that accurately reflects the system's energy dynamics and a force network that correctly represents the influence of external controls.

5.2. Evaluating Hamiltonian behavior

To evaluate the effect of the proposed EWB regularization, we introduce two novel metrics specifically designed to assess Hamiltonian behavior in the learned models, complementing the standard reconstruction metrics defined in Section 3. These metrics serve a diagnostic purpose, quantifying the physical nature of learned dynamics by evaluating energy conservation and force prediction behavior in a zero-control test setting. Since these metrics do not necessarily represent overall prediction accuracy, they are used to complement traditional reconstruction metrics to provide a better understanding of the learned dynamics model.

ZFEC. This metric is motivated by the principle that Hamiltonian systems are inherently energy-conserving in the absence of external forces (Eq. (2)). Therefore, a model that has effectively learned Hamiltonian dynamics should conserve energy when unrolled without external forces, even if trained on data with control inputs. We evaluate this behavior by unrolling the model on the zero-control test set while suppressing the force network A_θ , ensuring the dynamics evolve without external influence. We then measure the relative energy difference at each timestep using the learned Hamiltonian function:

$$\text{ZFEC} = \frac{1}{T} \sum_{t=0}^T \frac{|\mathcal{H}_\theta(\tilde{\mathbf{s}}_{t+1}) - \mathcal{H}_\theta(\tilde{\mathbf{s}}_t)|}{E + \kappa}, \quad (23)$$

Table 3

ZFEC, ZCEC and ZCFP measured on the zero-control test set. Lower is better for all metrics.

Environment	$\alpha = 0.0$			$\alpha = 1.0$			$\alpha = 1e^{-6} \rightarrow 1.0$		
	ZFEC	ZCEC	ZCFP	ZFEC	ZCEC	ZCFP	ZFEC	ZCEC	ZCFP
Pendulum	2.10e ⁶	6.13	1.63	3.23	1.55	0.80	7.04e ³	1.40	0.97
Reacher	2.10e ²	2.12	5.77	1.24	0.93	1.45	2.10	0.69	2.96
CartPole	3.05e ⁶	2.62	1.83	3.13e²	0.30	0.54	3.92e ²	0.26	0.32
Acrobot	1.09e ⁶	10.87	0.87	1.48e¹	5.16	0.45	1.11e ³	4.19	0.42
Cart2Pole	8.98e ⁵	14.28	1.01	5.28e⁴	3.40	0.33	3.50e ⁵	9.21	0.50
Cart3Pole	1.75e ⁵	8.22	0.47	2.00e⁴	8.55	0.33	4.53e ⁵	7.45	0.25
Acro3bot	9.65e ⁴	4.42	0.28	4.93e⁴	8.00	0.38	8.19e ⁴	4.89	0.32
Average	1.06e ⁶	6.95	1.69	1.75e⁴	3.98	0.61	1.28e ⁵	4.01	0.82

where E is the mean absolute latent energy taken over the entire test set. A low ZFEC score indicates energy conservation in the absence of external forces, suggesting the learned dynamics have effectively captured the Hamiltonian nature of the system. Conversely, a high ZFEC score implies that the learned dynamics rely on the force network to compensate for energy fluctuations, failing to capture the intrinsic conservation laws.

ZCFP. Complementing ZFEC, we introduce the ZCFP metric to evaluate whether the model has correctly decoupled the underlying Hamiltonian dynamics from the energy contributions of external forces. While ZFEC assesses the conservation properties of the learned Hamiltonian dynamics, ZCFP verifies whether the model correctly identifies that a zero-control input corresponds to zero applied force. We quantify this by measuring the average predicted latent force when unrolling the model on the zero-control test set:

$$\text{ZCFP} = \frac{1}{T} \sum_{t=0}^T \frac{|A_\theta(\tilde{s}_t, \mathbf{u}_t^0(\mathbf{x}_t))|}{P + \kappa}, \quad (24)$$

where $\mathbf{u}_t^0(\mathbf{x}_t)$ denotes the zero-control input at time t given state $\mathbf{x}_t$,⁴ and P is the mean absolute latent momentum $\tilde{p}$ taken over the entire test set. A low ZCFP score indicates that the model correctly associates zero action with zero latent force, whereas a high score suggests a failure to decouple the Hamiltonian dynamics from external forces.

Normalization constants. The normalization constants E and P are required to allow for comparison between experiments, since the model can learn arbitrary energy and coordinate scales, while a small constant $\kappa = 10^{-8}$ is used to avoid division by zero. To minimize the sensitivity of the metrics to this scaling effect, we propose to aggregate the normalization constants over the entire test set. In Appendix C we provide a sensitivity analysis of this normalization approach, comparing it with aggregation over trajectories and the unnormalized version of each metric on a Pendulum subject to known disturbances. We also present the normalization constants E and P for each experiment in Table C.1 to show that the resulting values are not merely a scaling effect due to small normalization values. We propose that ZFEC and ZCFP are always presented alongside their normalization factors to highlight outliers caused by small normalization constants.

5.3. Results

We evaluate the EWB model using two regularization schedules: a constant strength of $\alpha = 1.0$, and a schedule that increases linearly from $\alpha = 10^{-6}$ to $\alpha = 1.0$ over the first 150 epochs. Table 3 presents the results of the ZFEC and ZCFP metrics for the β -ACHGN++ model trained with and without the proposed EWB regularization. The table also presents

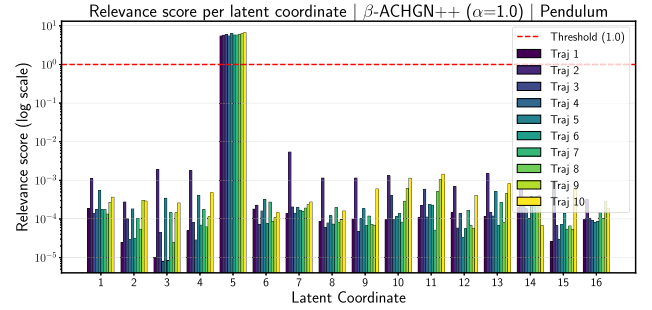


Fig. 2. Histogram of relevance scores for each latent coordinate pair in the ACHGN++ model trained on the Pendulum environment with EWB regularization ($\alpha = 1.0$) for the 10 long test trajectories. The model consistently uses only a single coordinate pair (pair 5) to represent the system dynamics. The dashed line indicates the threshold used to identify active coordinates.

a variant of the ZFEC metric which does not suppress the force network, referred to as ZCEC, providing a more realistic view of the behavior of the model during deployment. Both regularization strategies significantly improve the Hamiltonian behavior of the learned dynamics, as evidenced by substantial reductions in ZFEC, ZCEC and ZCFP scores across most environments, with the most complex environments seeing smaller differences. Notably, the constant regularization strength shows the least reliance on the force network to correct energy drift, as indicated by the lower ZCFP scores and the smaller gap between ZFEC and ZCEC. However, these metrics alone are insufficient to assess the quality of the learned dynamics, as trivial solutions could achieve low scores while failing to capture the system dynamics (e.g., the force network predicting zero force at all times).

Table 4 compares the MSE and VPT reconstruction metrics of the regularized variants with the baseline on the normal test set as well as on the zero-control test set. The results indicate minor improvements in MSE scores on the normal test set, with a larger improvement in long-term prediction capabilities indicated by the VPT scores. More notably, the results show a strong improvement in extrapolation and long-term prediction on most environments in the zero-control test set for both regularization schedules, matching the lower ZFEC, ZCEC and ZCFP scores of the regularized variants. These findings indicate that, while accurate short-term reconstruction is achievable without strictly enforcing Hamiltonian behavior, imposing the energy-work balance through regularization significantly enhances the model's long-term predictive capability in both in-distribution and out-of-distribution regimes. The minimal improvements achieved in the most complex environments once again indicate that these environments are not effectively represented by both the regularized and unregularized variants of ACHGN++. The consistent difficulty to predict the dynamics of the Cart3Pole and Acro3bot environments shows that high-complexity dynamics and strong underactuation represent possible failure modes for all architectures evaluated in this work.

⁴ Note that in the systems considered in this work, the zero-control input is independent of time and state: $\mathbf{u}_t^0(\mathbf{x}_t) = \mathbf{u}^0$.

Table 4

Reconstruction (Rec) and extrapolation (Ext) refer to the normalized MSE measured across the first 60 and subsequent 60 timesteps on the test set, respectively. VPT refers to the number of predicted timesteps before the normalized MSE exceeds a threshold λ of 0.9. MSE values are reported as multiples of 10^{-2} . Z-Rec, Z-Ext and Z-VPT metrics are calculated on the zero-control test set.

Environment	$\alpha = 0.0$						$\alpha = 1.0$						$\alpha = 1e^{-6} \rightarrow 1.0$					
	Rec	Ext	VPT	Z-Rec	Z-Ext	Z-VPT	Rec	Ext	VPT	Z-Rec	Z-Ext	Z-VPT	Rec	Ext	VPT	Z-Rec	Z-Ext	Z-VPT
Pendulum	0.16	2.67	404	5.74	40.06	11	0.16	2.42	557	9.20	20.42	35	0.24	2.52	548	11.11	30.74	15
Reacher	0.92	3.43	282	2.19	16.75	18	0.83	3.72	349	0.23	1.24	912	0.91	3.41	336	0.26	1.23	840
CartPole	0.14	0.73	710	4.03	58.50	27	0.16	0.61	742	1.45	21.53	58	0.14	0.74	752	1.09	20.67	43
Acrobot	2.03	27.07	100	6.12	57.50	5	2.21	28.00	105	11.81	41.23	23	2.18	29.38	78	11.02	41.56	2
Cart2Pole	0.97	13.94	100	5.88	157.94	7	0.96	11.86	111	7.35	23.03	20	1.72	14.68	113	6.08	28.00	11
Cart3Pole	17.39	93.59	2	19.83	139.44	0	19.31	94.69	10	27.25	128.74	0	18.48	91.41	0	21.86	163.28	0
Acro3bot	25.61	113.34	4	32.13	196.77	0	21.96	112.31	12	27.02	146.00	0	26.95	115.18	7	28.33	138.39	0
Average	6.75	36.40	229	10.84	95.28	10	6.52	36.23	269	12.04	54.60	150	7.23	36.76	262	11.39	60.55	130

5.4. Interpreting ACHGN

This section highlights the potential of ACHGN++ architectures to yield interpretable latent spaces that closely approximate canonical phase spaces. We demonstrate this by visualizing the latent space of a model trained on the Pendulum environment. Although the model has 16 latent coordinate pairs available, we find that it consistently utilizes only a single pair to represent the system dynamics. We identify this active coordinate pair by thresholding the KL divergence between the prior and posterior distributions of each latent dimension, summing the KL values per coordinate pair to define a relevance score:

$$\text{Relevance}(i) = \text{KL}[\tilde{\mathbf{q}}_i \parallel p(\mathbf{z})] + \text{KL}[\tilde{\mathbf{p}}_i \parallel p(\mathbf{z})], \quad (25)$$

where $\tilde{\mathbf{q}}_i$ and $\tilde{\mathbf{p}}_i$ are the latent position and momentum distributions for coordinate pair i , respectively, and $p(\mathbf{z})$ is the prior distribution. Latent variables with a low KL divergence are similar to the unit normal prior, indicating high standard deviations meaning they are likely treated as noise by the model. Fig. 2 visualizes these relevance scores for the β -ACHGN++ model with $\alpha = 1.0$ for the 10 long test trajectories on the Pendulum environment. The histogram clearly shows that a single coordinate pair (index 5) is consistently active across different trajectories, exhibiting a significantly higher relevance score than the others. Fig. 3 compares the learned phase space of this active coordinate pair with the true phase space of the pendulum for both controlled and zero-control trajectories. The learned phase space closely resembles the true phase space, preserving volume in energy-conserving settings and accurately reflecting energy changes under control inputs. In the controlled trajectory, an energy increase occurs halfway through the rollout, manifested as an increase in phase-space volume; this behavior is accurately captured in the learned phase space. The zero-control trajectory exhibits a closed orbit in the true phase space, while the learned phase space shows only a slight deviation from a closed orbit, indicating minimal energy drift.

Table 5 lists the number of active latents after training per environment, comparing the base ACHGN model with β -ACHGN++ models trained with and without EWB regularization. We observe that the architectural improvements in ACHGN++ lead to a significant reduction in the number of active latents across all but the most complex environments. Adding regularization further reduces the number of active latents in most environments, with the Cart3Pole and Acro3bot environments not showing any reduction in active latents for all variants. Notably, ACHGN++ models often require more active latents than the theoretical minimum, with the number increasing rapidly with environment complexity. This is likely caused by the training objective optimizing solely for reconstruction accuracy, with only the KL term in Eq. (13) discouraging unnecessary information usage. Furthermore, the force network A_θ is conditioned on the full latent state and can impact each coordinate pair independently, possibly causing dormant coordinate pairs to become active over the course of an unrolled trajectory.

Table 5

Number of relevant latent coordinate pairs used by the learned models. $\alpha = 0.0$ and $\alpha = 1.0$ refer to β -ACHGN++ models trained without and with EWB regularization, respectively. A coordinate pair is considered relevant if its relevance score (as defined in Eq. (25)) exceeds a threshold of 1.0. GT refers to the minimal number of latent coordinate pairs required to represent the true state of the environment.

Environment	GT	ACHGN	$\alpha = 0.0$	$\alpha = 1.0$
Pendulum	1	32	2	1
Reacher	2	32	5	4
CartPole	2	32	4	4
Acrobot	2	32	12	8
Cart2Pole	3	32	10	8
Cart3Pole	4	32	32	32
Acro3bot	3	32	32	32

6. Limitations and future work

The extended Hamiltonian dynamics presented in Eq. (14) assume control-affinity, an assumption which extends to the regularization proposed in Eq. (22). While many physical systems adhere to this constraint, systems subject to non-affine forces cannot be effectively captured by Eq. (14) and are therefore not subject to the regularization as presented in Section 5.1. Furthermore, the extended Hamiltonian dynamics do not account for aspects such as friction and damping which are present in most real-world settings and significantly increase the complexity of disambiguating external forces from the underlying environment dynamics. Since the experiments presented in this work are performed on systems which are not subject to these phenomena, they present a relatively idealized setting and do not accurately represent real-world systems. Promising avenues for future work can aim to incorporate existing approaches extending Hamiltonian Neural Networks to the pseudo-Hamiltonian formulation into ACHGN++ to increase the practical applicability of the architecture [18,19].

The experiments conducted in this work evaluate the proposed architecture and regularization on low-dimensional, non-canonical input spaces, which are not representative of image or other high-dimensional sensors. However, given the autoencoder architecture and previous works successfully learning Hamiltonian dynamics from images [13,31], learning Hamiltonian dynamics subject to external control from high-dimensional inputs is a promising avenue for further research and experimentation.

The strong performance degradation observed in the most complex and underactuated environments for all architectures evaluated in this work suggest a more fundamental limitation of the current formulation of latent physics-informed learning approaches. A promising direction for future work is to investigate the underlying causes of this performance degradation and to explore potential modifications to the architecture, training procedure, or regularization to mitigate this issue.

More broadly, this work occupies an intermediate position in the trade-off between model flexibility and physical grounding in

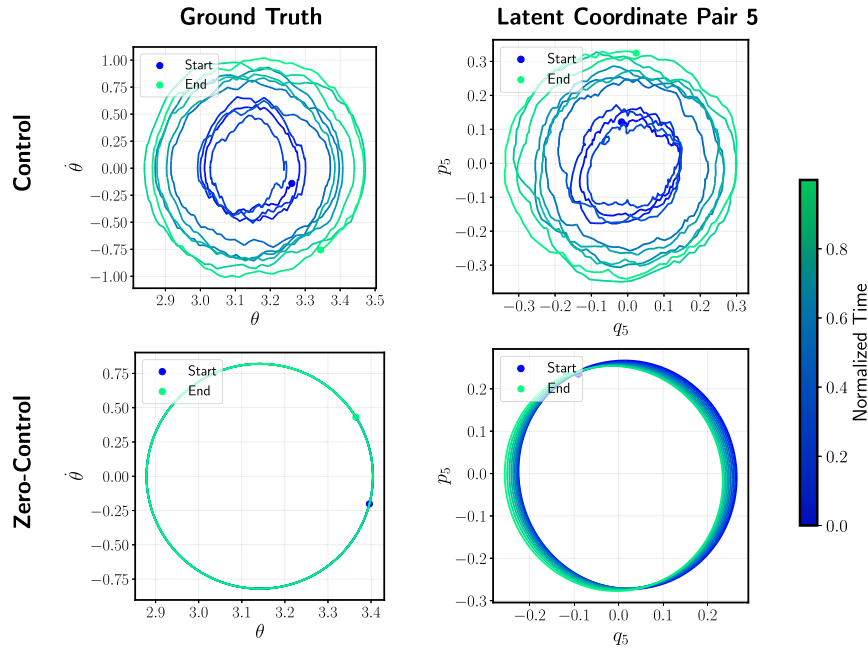


Fig. 3. Comparing the learned phase space of active latent coordinate pair 5 with the true phase space of the Pendulum environment under both controlled and zero-control trajectories.

physics-informed learning. Compared to strongly constrained approaches such as PINNs [5,7], ACHGN++ retains greater representational flexibility by not enforcing full governing-equation residuals. However, this also means physical consistency is only partially enforced through Hamiltonian structure and EWB regularization. As a result, while we observe improved long-horizon and zero-control behavior, stronger constraints may be required in settings with dissipation, contact, or other effects that violate the assumptions of the current formulation.

7. Conclusion

This work presents ACHGN++, an extension to the ACHGN architecture that combines several architectural and training modifications to enhance the learning of Hamiltonian dynamics under external control. We compare this architecture with established baselines across a suite of seven dissipation-free controlled dynamical systems and demonstrate its superior performance in both short-term reconstruction and long-term prediction on all but the most complex systems. We further improve the learning process by introducing a novel regularization technique based on the energy-work balance which leads to improved Hamiltonian behavior of the learned dynamics. To assess this behavior, we propose two new metrics, ZFEC and ZCFP, which quantify the energy conservation properties and force prediction accuracy of the learned models in zero-control scenarios. Our results indicate that ACHGN++ models trained with the proposed regularization not only exhibit enhanced Hamiltonian characteristics but also achieve better long-term prediction performance. Finally, we demonstrate that ACHGN++ learns interpretable latent spaces resembling canonical coordinates, with interpretability inversely correlated to system complexity: simple systems yield highly structured latent representations, while complex dynamics result in less interpretable encodings.

CRedit authorship contribution statement

Arne Troch: Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization; **Kevin Mets:** Writing – review & editing, Supervision, Funding acquisition; **Siegfried Mercelis:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Arne Troch reports financial support was provided by Research Foundation Flanders. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Research Foundation Flanders (FWO) under Grant Number 1SHAH24N. This research received funding from the Flemish Government (Flanders AI Research Program).

Data availability

I have shared the link to my data at the Attach File step

Appendix A. Environments

All angle elements of the environments are in radians and are randomly initialized within the range $[\pi, 2\pi]$. The CartPole environments have an additional positional element (the position of the cart) which is randomly initialized within the range $[0, 0.5]$. While these ranges do not fully cover the state space (e.g., the angles can reach values within $[0, 2\pi]$), sampling subsequences from longer random rollouts ensures that the training data covers a sufficient part of the state space. Fig. A.1 demonstrates this by visualizing the state space coverage of the training dataset for the Pendulum, CartPole and Acrobot environments. We describe the state space of each environment below (note that we describe the systems in canonical coordinates, not the observed states used for training). The observed states consist of the sine and cosine of each angle together with the (angular) velocities. For all environments, the convention is that $\theta = 0$ corresponds to the vertically upward position.⁵

⁵ The code for the environments can be found at: https://github.com/adi3e08/Physics_Informed_Model_Based_RL

Table A.1

Overview of the datasets used for training and evaluation. The training and test sets are subsampled from longer unrolled trajectories with random actions. The zero-control dataset is collected with $\mathbf{u} = \mathbf{0}$.

Dataset	Total Sequences	Sequence Length	Trajectory Length	Sequences per Trajectory
Training	50,000	64	300	10
Test	20,000	125	1000	5
Long Test	10	1000	1000	1
Zero-Control Test	5000	125	1000	5

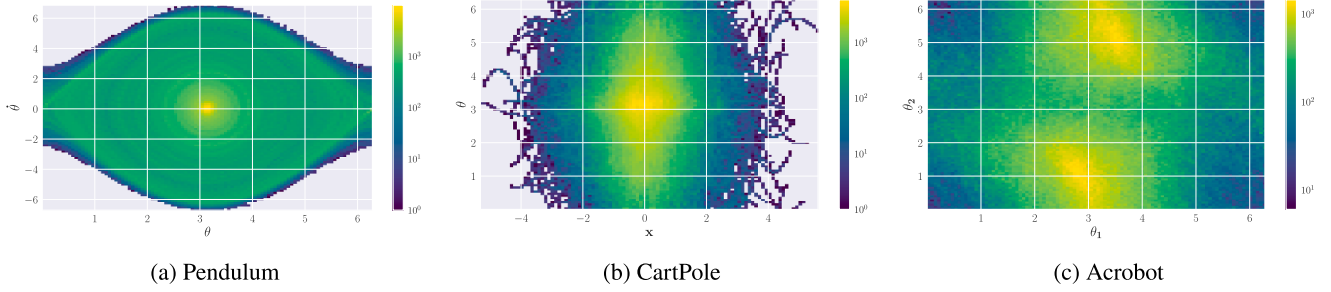


Fig. A.1. State space densities of the training dataset. The 2-dimensional state space of the Pendulum environment is fully visualized in (a). The CartPole and Acrobot environments both have 4D state spaces consisting of positional and velocity elements. For visualization purposes we only visualize the positional elements in (b) and (c).

- **Pendulum:** $\Delta t = 0.02$; $\mathbf{x} = [\theta, \dot{\theta}]$, where θ is the angle of the pendulum and $\dot{\theta}$ is its angular velocity.
- **CartPole:** $\Delta t = 0.01$; $\mathbf{x} = [x, \theta, \dot{x}, \dot{\theta}]$, where x is the position of the cart, $\dot{x}$ is its velocity, θ is the angle of the pole, and $\dot{\theta}$ is its angular velocity. Only the cart is actuated.
- **Acrobot:** $\Delta t = 0.01$; $\mathbf{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]$, where θ_1 and θ_2 are the angles of the two links, and $\dot{\theta}_1$ and $\dot{\theta}_2$ are their respective angular velocities. Only the second joint is actuated.
- **Reacher:** $\Delta t = 0.02$; $\mathbf{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]$, where θ_1 and θ_2 are the angles of the two links, and $\dot{\theta}_1$ and $\dot{\theta}_2$ are their respective angular velocities. Note that this environment is different from the Acrobot in that it is fully actuated (both joints can be controlled) and is manipulated in the horizontal plane rather than the vertical plane.
- **Cart2Pole:** $\Delta t = 0.01$; $\mathbf{x} = [x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2]$, where x is the position of the cart, $\dot{x}$ is its velocity, θ_1 and θ_2 are the angles of the two poles, and $\dot{\theta}_1$ and $\dot{\theta}_2$ are their respective angular velocities. Only the cart is actuated.
- **Cart3Pole:** $\Delta t = 0.01$; $\mathbf{x} = [x, \theta_1, \theta_2, \theta_3, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]$, where x is the position of the cart, $\dot{x}$ is its velocity, θ_1 , θ_2 and θ_3 are the angles of the three poles, and $\dot{\theta}_1$, $\dot{\theta}_2$ and $\dot{\theta}_3$ are their respective angular velocities. The cart and the third pole are actuated.
- **Acro3bot:** $\Delta t = 0.01$; $\mathbf{x} = [\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]$, where θ_1 , θ_2 and θ_3 are the angles of the three links, and $\dot{\theta}_1$, $\dot{\theta}_2$ and $\dot{\theta}_3$ are their respective angular velocities. Only the first and third joints are actuated.

Appendix B. Energy-work balance

Starting from the time derivative of the Hamiltonian:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{q}} \cdot \frac{d\mathbf{q}}{dt} + \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \cdot \frac{d\mathbf{p}}{dt} \quad (\text{B.1})$$

Substituting the dynamics from Eq. (14):

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{q}} \cdot \frac{\partial \mathcal{H}}{\partial \mathbf{p}} + \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \cdot \left(-\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \mathbf{g}(\mathbf{q})\mathbf{u} \right) \quad (\text{B.2})$$

The first two terms cancel out, yielding:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \cdot \mathbf{g}(\mathbf{q})\mathbf{u} \quad (\text{B.3})$$

Appendix C. Validation of ZFEC and ZCFP

We create various synthetic disturbances in the Pendulum environment which aim to validate the behavior of the proposed metrics and compare this behavior with MSE and unnormalized trends. We create three disturbances, each designed to target specific aspects of the learned dynamics and the corresponding metrics:

1. **Energy Drift** is achieved by adding force to the pendulum in order to move the system away from equilibrium. Three versions of energy drift are investigated by adjusting the direction of the added force. A monotonically increasing energy profile is achieved by adding force matching the direction of the pendulum swing, a monotonically decreasing energy profile is achieved by adding force in the opposite direction of the pendulum swing and a varied energy profile is achieved by choosing either direction randomly with equal odds. This disturbance aims to demonstrate the behavior of the ZFEC metric.

Table C.1

Regularization constants for ZFEC, ZCEC and ZCFP metrics shown in Table 3.

Environment	$\alpha = 0.0$			$\alpha = 1.0$			$\alpha = 1e^{-6} \rightarrow 1.0$		
	E-ZFEC	E-ZCEC	P-ZCFP	E-ZFEC	E-ZCEC	P-ZCFP	E-ZFEC	E-ZCEC	P-ZCFP
Pendulum	2.90	1.43	0.54	0.99	0.89	0.46	1.09	1.14	0.42
Reacher	1.53	1.34	0.43	1.03	1.13	0.35	1.02	1.13	0.42
CartPole	0.87	1.10	0.36	0.94	0.95	0.33	1.11	1.12	0.34
Acrobot	2.11	2.08	0.83	1.01	1.09	0.76	1.18	1.62	0.81
Cart2Pole	3.73	2.85	0.94	1.15	1.07	0.84	1.08	1.06	0.82
Cart3Pole	6.34	5.96	1.93	1.62	1.87	1.69	0.52	1.45	1.67
Acro3bot	6.47	5.92	2.01	1.43	1.58	1.32	2.16	2.43	1.21

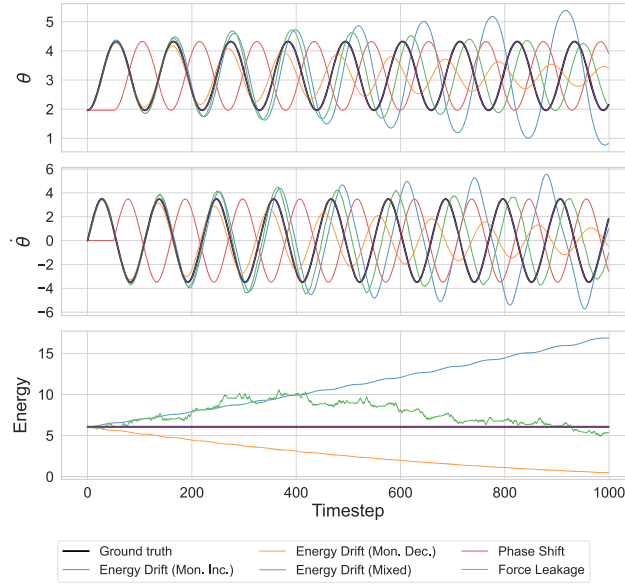


Fig. C.1. Example trajectory of each disturbance with maximum intensity compared to a ground truth trajectory. Force leakage disturbance overlaps with ground truth since this disturbance does not affect the system dynamics directly.

2. **Phase Shift** is achieved by arbitrarily shifting the trajectory in time, effectively creating a phase shift in the oscillation of the pendulum. This disturbance should heavily affect the MSE metric while not impacting both ZFEC and ZCFP metrics, since there is no shift in energy and there are no external forces.
3. **Force Leakage** is simulated by keeping the force in the Pendulum system at zero, but falsely presenting a control input to the metric. This disturbance aims to demonstrate the behavior of the ZCFP metric.

The intensity of each disturbance is scaled between zero and one, and example trajectories of each disturbance with maximum intensity are shown on Fig. C.1. For each intensity value, we create 500 trajectories of length 1000 with random initial conditions as described in Appendix A and plot the mean and standard deviation of each metric. In our analysis, we compare trajectory and test-set normalization with the unnormalized metric and provide a base MSE metric to indicate overall deviation from ground truth.

C.1. Sensitivity analysis of ZFEC

Fig. C.2 shows a sensitivity analysis of the ZFEC metric on the energy drift and phase shift disturbances over ten intensity values. The results demonstrate that normalization using aggregate test-set statistics most closely tracks the unnormalized metric while maintaining low standard deviation. In contrast, trajectory-level normalization exhibits inconsistent behavior with substantially larger standard deviations that do not align with the unnormalized trend.

C.2. Sensitivity analysis of ZCFP

Fig. C.3 shows a sensitivity analysis of the ZCFP metric on the force leakage and phase shift disturbances over ten intensity values. We provide three versions of force leakage: a smooth profile proportional to $\tanh(\mathbf{p})$, a noisy profile which adds Gaussian noise and a discrete profile with varying amplitude and frequency per trajectory. We compare trajectory and test-set normalization with the unnormalized version and provide a base MSE metric to indicate overall deviation from ground truth. The results follow a similar trend as the ZFEC sensitivity analysis, with test-set normalization closely tracking the unnormalized metric while maintaining low standard deviation, and trajectory-level normalization exhibiting inconsistent behavior with substantially larger standard deviations that do not align with the unnormalized trend.

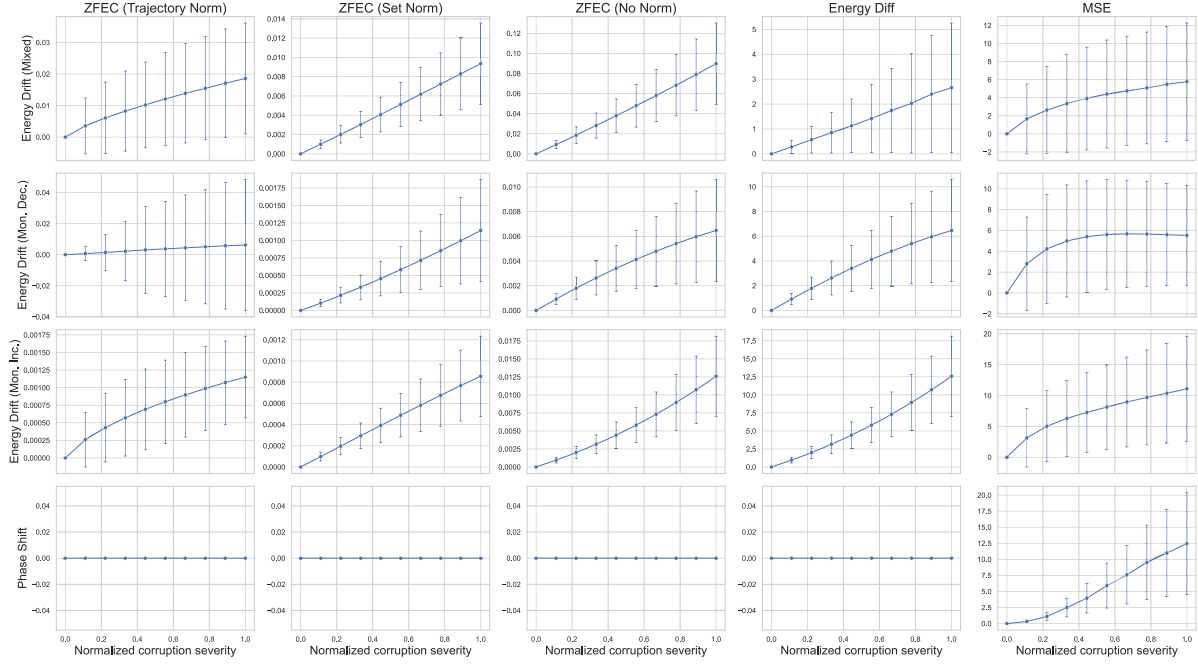


Fig. C.2. Sensitivity analysis of various normalization approaches for the ZFEC metric.

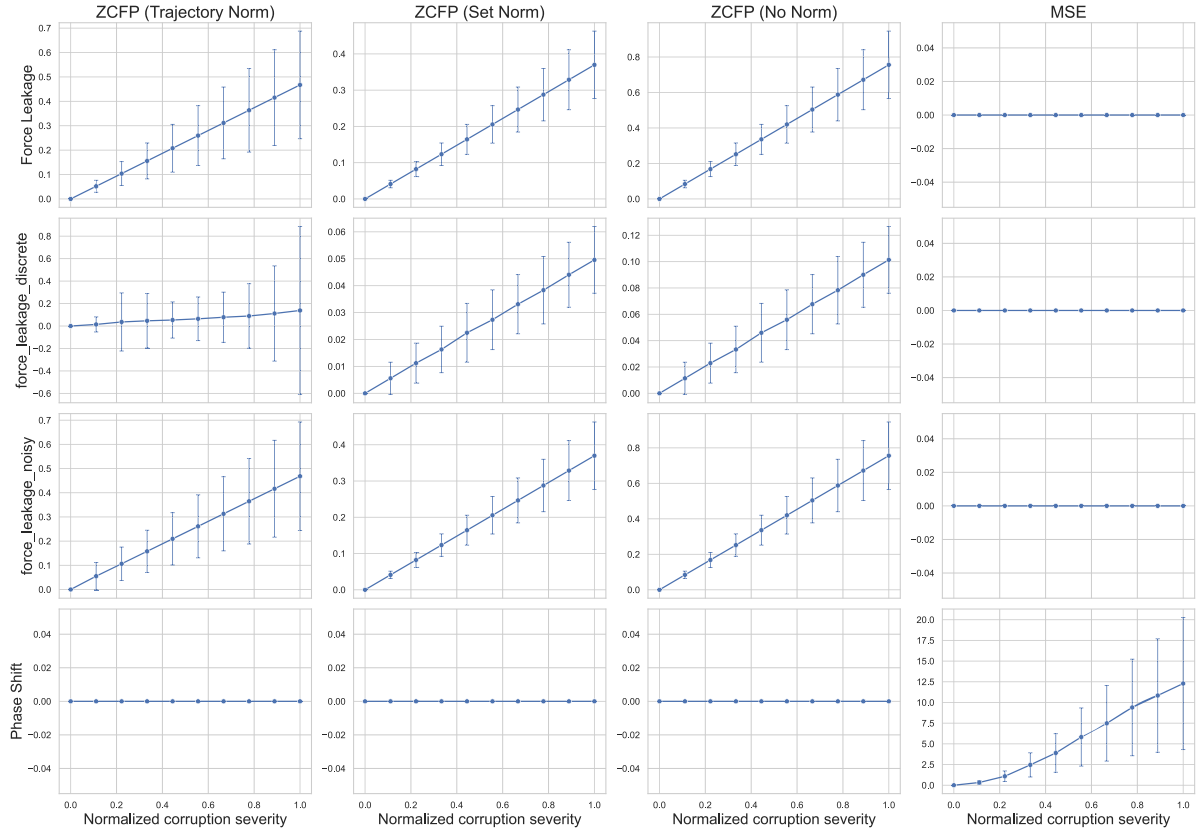
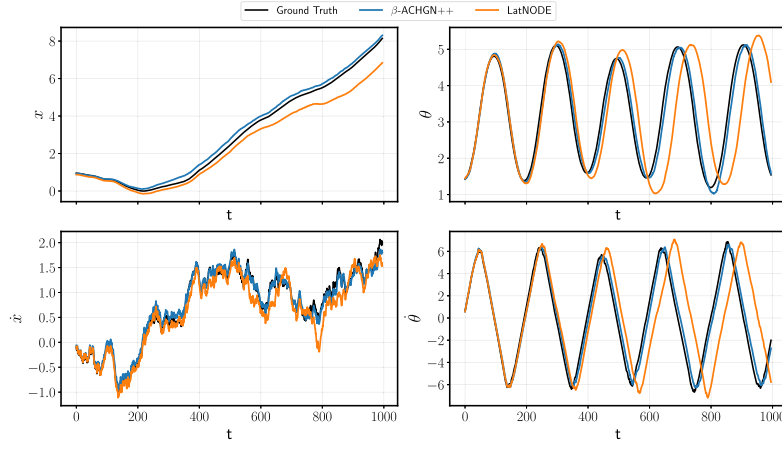
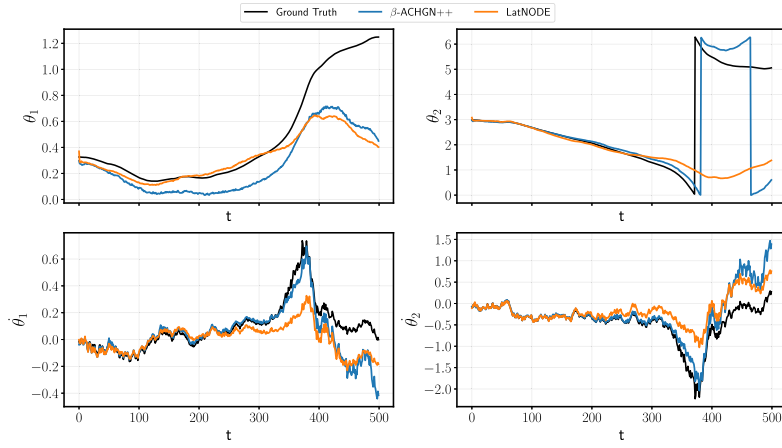


Fig. C.3. Sensitivity analysis of various normalization approaches for the ZCFP metric.

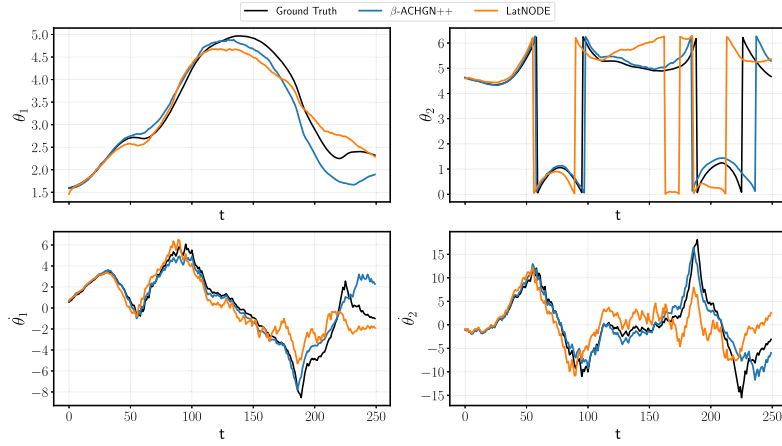
Appendix D. Extra trajectories



(a) CartPole



(b) Reacher



(c) Acrobot

Fig. D.1. Visualization of ground truth and predicted trajectories for controlled trajectories on the CartPole, Reacher and Acrobot systems. Note the truncated x-axis range for Reacher and Acrobot trajectories.

Appendix E. Evaluation results including inter-seed standard deviations

Note that the reported results are presented over 3 random seeds, with the mean and standard deviation of each metric across seeds reported in the tables below. While the standard deviations indicate low variability across seeds for almost all experiments, complete statistical significance can not be claimed using 3 random seeds (Tables E.1–E.15).

E.1. Architectural experiments

Table E.1

Reconstruction (**Rec.**) performance of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	ACHGN	β -VAE	Leap Frog	No Transform	Separable	Learned Δt
Pendulum	0.07 ± 0.00	0.71 ± 0.02	0.04 ± 0.00	0.07 ± 0.01	0.11 ± 0.02	0.06 ± 0.00
Reacher	0.12 ± 0.01	0.82 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.10 ± 0.02	0.10 ± 0.01
CartPole	0.03 ± 0.00	0.22 ± 0.03	0.09 ± 0.01	0.01 ± 0.00	0.03 ± 0.00	0.06 ± 0.00
Acrobot	3.05 ± 0.07	2.75 ± 0.06	2.65 ± 0.06	2.66 ± 0.05	2.53 ± 0.06	2.85 ± 0.06
Cart2Pole	2.07 ± 0.04	2.43 ± 0.05	2.65 ± 0.05	1.45 ± 0.03	1.77 ± 0.04	2.86 ± 0.06
Cart3Pole	24.89 ± 0.54	16.97 ± 0.37	26.22 ± 0.57	19.30 ± 0.40	63.11 ± 1.30	24.72 ± 0.51
Acro3bot	81.75 ± 1.68	70.88 ± 1.46	77.17 ± 1.69	24.54 ± 0.54	62.89 ± 1.38	24.07 ± 0.50

Table E.2

Extrapolation (**Ext.**) performance of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	ACHGN	β -VAE	Leap Frog	No Transform	Separable	Learned Δt
Pendulum	10.14 ± 0.22	13.48 ± 0.30	8.48 ± 0.19	6.92 ± 0.15	9.86 ± 0.22	4.32 ± 0.09
Reacher	0.78 ± 0.02	4.86 ± 0.10	0.78 ± 0.02	0.68 ± 0.01	1.82 ± 0.04	0.76 ± 0.02
CartPole	0.68 ± 0.01	2.48 ± 0.05	0.69 ± 0.02	0.49 ± 0.01	1.29 ± 0.03	1.16 ± 0.02
Acrobot	65.64 ± 1.35	62.77 ± 1.29	68.53 ± 1.50	72.59 ± 1.50	61.25 ± 1.26	50.94 ± 1.11
Cart2Pole	35.17 ± 0.72	36.15 ± 0.75	34.28 ± 0.71	33.89 ± 0.70	33.45 ± 0.69	24.01 ± 0.49
Cart3Pole	125.34 ± 2.58	105.19 ± 2.30	124.00 ± 2.56	108.45 ± 2.24	122.36 ± 2.68	99.66 ± 2.05
Acro3bot	143.53 ± 2.96	124.89 ± 2.73	135.29 ± 2.96	129.75 ± 2.84	125.31 ± 2.74	111.95 ± 2.45

Table E.3

Long-term prediction (**VPT**) performance ($\lambda = 0.9$) of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation.

Environment	ACHGN	β -VAE	Leap Frog	No Transform	Separable	Learned Δt
Pendulum	133 ± 3	102 ± 2	128 ± 3	136 ± 3	130 ± 3	140 ± 3
Reacher	210 ± 5	194 ± 4	226 ± 5	242 ± 5	231 ± 5	218 ± 5
CartPole	160 ± 3	149 ± 3	191 ± 4	181 ± 4	180 ± 4	163 ± 3
Acrobot	95 ± 2	54 ± 1	65 ± 1	75 ± 2	93 ± 2	106 ± 2
Cart2Pole	87 ± 2	80 ± 2	98 ± 2	96 ± 2	98 ± 2	94 ± 2
Cart3Pole	4 ± 1	10 ± 2	0 ± 0	8 ± 3	0 ± 0	0 ± 0
Acro3bot	0 ± 0	0 ± 0	0 ± 0	7 ± 4	0 ± 0	0 ± 0

E.2. Baseline experiments

Table E.4

Reconstruction (**Rec.**) performance of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	ACHGN	β -ACHGN + +	γ -ACHGN + +	LatNODE	RGN	HGN
Pendulum	0.07 ± 0.00	0.16 ± 0.02	0.04 ± 0.00	0.29 ± 0.01	0.29 ± 0.01	1.19 ± 0.03
Reacher	0.12 ± 0.01	0.92 ± 0.02	0.72 ± 0.01	0.94 ± 0.02	0.84 ± 0.02	6.25 ± 0.13
CartPole	0.03 ± 0.00	0.14 ± 0.03	0.12 ± 0.01	0.20 ± 0.02	0.19 ± 0.03	2.75 ± 0.06
Acrobot	3.05 ± 0.07	2.03 ± 0.04	2.65 ± 0.06	2.96 ± 0.06	4.00 ± 0.09	35.01 ± 0.77
Cart2Pole	2.07 ± 0.04	0.97 ± 0.02	1.78 ± 0.04	1.24 ± 0.03	1.18 ± 0.03	3.77 ± 0.08
Cart3Pole	24.89 ± 0.54	17.39 ± 0.36	16.65 ± 0.36	15.54 ± 0.34	25.10 ± 0.55	31.47 ± 0.69
Acro3bot	81.75 ± 1.68	25.61 ± 0.56	26.39 ± 0.58	28.12 ± 0.58	27.00 ± 0.56	57.58 ± 1.26

Table E.5

Extrapolation (**Ext.**) performance of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	ACHGN	β -ACHGN + +	γ -ACHGN + +	LatNODE	RGN	HGN
Pendulum	10.14 ± 0.22	2.67 ± 0.05	4.16 ± 0.09	2.41 ± 0.05	3.20 ± 0.07	19.24 ± 0.42
Reacher	0.78 ± 0.02	3.43 ± 0.08	0.48 ± 0.01	3.65 ± 0.08	3.71 ± 0.08	26.84 ± 0.55
CartPole	0.68 ± 0.01	0.73 ± 0.02	0.18 ± 0.00	0.67 ± 0.01	0.89 ± 0.02	13.42 ± 0.29
Acrobot	65.64 ± 1.35	27.07 ± 0.56	48.19 ± 0.99	23.97 ± 0.52	45.51 ± 0.94	102.98 ± 2.25
Cart2Pole	35.17 ± 0.72	13.94 ± 0.29	21.95 ± 0.45	13.40 ± 0.28	13.52 ± 0.30	42.56 ± 0.93
Cart3Pole	125.34 ± 2.58	93.59 ± 1.93	98.09 ± 2.15	86.31 ± 1.89	90.91 ± 1.87	114.07 ± 2.35
Acro3bot	143.53 ± 2.96	113.34 ± 2.48	110.54 ± 2.42	109.83 ± 2.40	109.84 ± 2.26	123.30 ± 2.70

Table E.6

Long-term prediction (VPT) performance ($\lambda = 0.9$) of different architectural variants of ACHGN, averaged across 3 random seeds with standard deviation.

Environment	ACHGN	β -ACHGN++	γ -ACHGN++	LatNODE	RGN	HGN
Pendulum	133 $\pm$ 3	404 $\pm$ 9	113 $\pm$ 2	369 $\pm$ 8	382 $\pm$ 8	107 $\pm$ 2
Reacher	210 $\pm$ 5	282 $\pm$ 6	206 $\pm$ 4	256 $\pm$ 6	312 $\pm$ 7	54 $\pm$ 1
CartPole	160 $\pm$ 3	710 $\pm$ 15	224 $\pm$ 5	517 $\pm$ 11	409 $\pm$ 9	58 $\pm$ 1
Acrobot	95 $\pm$ 2	100 $\pm$ 2	85 $\pm$ 2	89 $\pm$ 2	45 $\pm$ 1	9 $\pm$ 0
Cart2Pole	87 $\pm$ 2	100 $\pm$ 2	103 $\pm$ 2	104 $\pm$ 2	81 $\pm$ 2	52 $\pm$ 1
Cart3Pole	4 $\pm$ 1	2 $\pm$ 1	14 $\pm$ 2	8 $\pm$ 3	0 $\pm$ 0	0 $\pm$ 0
Acro3bot	0 $\pm$ 0	4 $\pm$ 2	9 $\pm$ 1	7 $\pm$ 4	0 $\pm$ 0	0 $\pm$ 0

E.3. Physics-based metrics of regularization experiments

Table E.7

ZFEC of various EWB regularization schedules, measured on the zero-control test set, averaged across three random seeds with standard deviation.

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	2.10e ⁶ $\pm$ 4.33e ⁴	3.23 $\pm$ 0.07	7.04e ³ $\pm$ 1.54e ²
Reacher	2.10e ² $\pm$ 4.61	1.24 $\pm$ 0.03	2.10 $\pm$ 0.04
CartPole	3.05e ⁶ $\pm$ 6.28e ⁴	3.13e ² $\pm$ 6.64	3.92e ² $\pm$ 8.09
Acrobot	1.09e ⁶ $\pm$ 2.25e ⁴	1.48e ¹ $\pm$ 0.13	1.11e ³ $\pm$ 2.29e ¹
Cart2Pole	8.98e ⁵ $\pm$ 1.97e ⁴	5.28e ⁴ $\pm$ 1.12e ³	3.50e ⁵ $\pm$ 7.66e ³
Cart3Pole	1.75e ⁵ $\pm$ 3.61e ³	2.00e ⁴ $\pm$ 4.25e ²	4.53e ⁵ $\pm$ 9.34e ³
Acro3bot	9.65e ⁴ $\pm$ 2.11e ³	4.93e ⁴ $\pm$ 1.05e ³	8.19e ⁴ $\pm$ 1.79e ³

Table E.8

ZCEC of various EWB regularization schedules, measured on the zero-control test set, averaged across three random seeds with standard deviation.

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	6.13 $\pm$ 0.13	1.55 $\pm$ 0.03	1.40 $\pm$ 0.03
Reacher	2.12 $\pm$ 0.05	0.93 $\pm$ 0.02	0.69 $\pm$ 0.01
CartPole	2.62 $\pm$ 0.05	0.30 $\pm$ 0.01	0.26 $\pm$ 0.01
Acrobot	10.87 $\pm$ 0.24	5.16 $\pm$ 0.11	4.19 $\pm$ 0.09
Cart2Pole	14.28 $\pm$ 0.29	3.40 $\pm$ 0.07	9.21 $\pm$ 0.19
Cart3Pole	8.22 $\pm$ 0.17	8.55 $\pm$ 0.18	7.45 $\pm$ 0.16
Acro3bot	4.42 $\pm$ 0.10	8.00 $\pm$ 0.16	4.89 $\pm$ 0.10

Table E.9

ZCFP of various EWB regularization schedules, measured on the zero-control test set, averaged across three random seeds with standard deviation.

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	1.63 $\pm$ 0.04	0.80 $\pm$ 0.02	0.97 $\pm$ 0.02
Reacher	5.77 $\pm$ 0.13	1.45 $\pm$ 0.03	2.96 $\pm$ 0.06
CartPole	1.83 $\pm$ 0.04	0.54 $\pm$ 0.01	0.32 $\pm$ 0.01
Acrobot	0.87 $\pm$ 0.02	0.45 $\pm$ 0.01	0.42 $\pm$ 0.01
Cart2Pole	1.01 $\pm$ 0.02	0.33 $\pm$ 0.01	0.50 $\pm$ 0.01
Cart3Pole	0.47 $\pm$ 0.01	0.33 $\pm$ 0.01	0.25 $\pm$ 0.01
Acro3bot	0.28 $\pm$ 0.01	0.38 $\pm$ 0.01	0.32 $\pm$ 0.01

E.4. Reconstruction metrics of regularization experiments

Table E.10

Reconstruction (**Rec.**) performance of various EWB regularization schedules, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	0.16 ± 0.02	0.16 ± 0.00	0.24 ± 0.00
Reacher	0.92 ± 0.02	0.83 ± 0.02	0.91 ± 0.02
CartPole	0.14 ± 0.03	0.16 ± 0.00	0.14 ± 0.00
Acrobot	2.03 ± 0.04	2.21 ± 0.05	2.18 ± 0.05
Cart2Pole	0.97 ± 0.02	0.96 ± 0.02	1.72 ± 0.04
Cart3Pole	17.39 ± 0.36	19.31 ± 0.42	18.48 ± 0.38
Acro3bot	25.61 ± 0.56	21.96 ± 0.48	26.95 ± 0.59

Table E.11

Extrapolation (**Ext.**) performance for various EWB regularization schedules, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	2.67 ± 0.05	2.42 ± 0.05	2.52 ± 0.06
Reacher	3.43 ± 0.08	3.72 ± 0.08	3.41 ± 0.07
CartPole	0.73 ± 0.02	0.61 ± 0.01	0.74 ± 0.02
Acrobot	27.07 ± 0.56	28.00 ± 0.58	29.38 ± 0.61
Cart2Pole	13.94 ± 0.29	11.86 ± 0.26	14.68 ± 0.30
Cart3Pole	93.59 ± 1.93	94.69 ± 1.99	91.41 ± 2.00
Acro3bot	113.34 ± 2.48	112.31 ± 2.40	115.18 ± 2.52

Table E.12

Long-term prediction (**VPT**) performance ($\lambda = 0.9$) of various EWB regularization schedules, averaged across 3 random seeds with standard deviation.

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	404 ± 9	557 ± 12	548 ± 11
Reacher	282 ± 6	349 ± 8	336 ± 7
CartPole	710 ± 15	742 ± 10	752 ± 12
Acrobot	100 ± 3	105 ± 2	78 ± 4
Cart2Pole	100 ± 5	111 ± 2	113 ± 1
Cart3Pole	2 ± 1	10 ± 3	0 ± 0
Acro3bot	4 ± 2	12 ± 2	7 ± 3

E.5. Zero-control reconstruction metrics of regularization experiments

Table E.13

Zero-control reconstruction (**Z-Rec.**) performance of various EWB regularization schedules on the zero-control test set, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	5.74 ± 0.12	9.20 ± 0.19	11.11 ± 0.24
Reacher	2.19 ± 0.05	0.23 ± 0.01	0.26 ± 0.01
CartPole	4.03 ± 0.08	1.45 ± 0.03	1.09 ± 0.02
Acrobot	6.12 ± 0.13	11.81 ± 0.24	11.02 ± 0.23
Cart2Pole	5.88 ± 0.12	7.35 ± 0.15	6.08 ± 0.13
Cart3Pole	19.83 ± 0.43	27.25 ± 0.56	21.86 ± 0.48
Acro3bot	32.13 ± 0.66	27.02 ± 0.56	28.33 ± 0.62

Table E.14

Zero-control extrapolation (Z-Ext.) performance for various EWB regularization schedules on the zero-control test set, averaged across 3 random seeds with standard deviation. Values are reported as multiples of 10^{-2} .

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	40.06 ± 0.88	20.42 ± 0.45	30.74 ± 0.67
Reacher	16.75 ± 0.37	1.24 ± 0.03	1.23 ± 0.03
CartPole	58.50 ± 1.28	21.53 ± 0.47	20.67 ± 0.43
Acrobot	57.50 ± 1.19	41.23 ± 0.90	41.56 ± 0.91
Cart2Pole	157.94 ± 3.46	23.03 ± 0.50	28.00 ± 0.61
Cart3Pole	139.44 ± 2.87	128.74 ± 2.65	163.28 ± 3.37
Acro3bot	196.77 ± 4.31	146.00 ± 3.20	138.39 ± 3.03

Table E.15

Zero-control long-term prediction (Z-VPT) performance ($\lambda = 0.9$) of various EWB regularization schedules on the zero-control test set, averaged across 3 random seeds with standard deviation.

Environment	$\alpha = 0.0$	$\alpha = 1.0$	$\alpha = 1e^{-6} \rightarrow 1.0$
Pendulum	11 ± 2	35 ± 3	15 ± 4
Reacher	18 ± 1	912 ± 21	840 ± 18
CartPole	27 ± 1	58 ± 1	43 ± 1
Acrobot	5 ± 3	23 ± 2	2 ± 1
Cart2Pole	7 ± 4	20 ± 3	11 ± 2
Cart3Pole	0 ± 0	0 ± 0	0 ± 0
Acro3bot	0 ± 0	0 ± 0	0 ± 0

Table E.16

Hyperparameters and network architectures consistent across experiments.

Hyperparameter	Value
Batch size	64
Latent space dimension	32
Inference steps	5
Target steps	60
Learning rate	1.0×10^{-3}
Epochs	200
q_θ - Number of units	128
q_θ - Number of layers	2
q_θ - Activation function	Leaky ReLU
H_θ - Number of units	256
H_θ - Number of layers	3
H_θ - Activation function	Swish
d_θ - Number of units	128
d_θ - Number of layers	2
d_θ - Activation function	Leaky ReLU
A_θ - Number of units	64
A_θ - Number of layers	2
A_θ - Activation function	ReLU

Table E.17

Computational cost comparison of ACHGN and baselines. **Param.** represents the total number of parameters in the model. **Inf.** represents the average and standard deviation inference time (in ms) on a batch 100 unrolled trajectories of length 1000 performed on an Apple M5 CPU. Note that the model architecture is kept constant for each environment, so differences in parameter count and inference time are entirely attributed to different input spaces.

Environment	ACHGN		β -ACHGN + +		LatNODE		RGN		HGN	
	Param.	Inf.	Param.	Inf.	Param.	Inf.	Param.	Inf.	Param.	Inf.
Pendulum	95K	71.78±0.72	159K	318.56±2.31	98K	76.02±7.66	98K	38.65±3.90	156K	317.67±3.10
Reacher	97K	77.30±1.21	161K	320.60±6.85	100K	83.22±9.68	100K	38.78±3.98	158K	321.49±2.23
CartPole	96K	72.10±4.80	160K	322.35±4.41	99K	76.74±8.56	99K	39.59±5.42	157K	321.72±5.91
Acrobot	97K	71.37±1.10	161K	322.73±8.03	100K	76.21±7.69	100K	38.67±3.91	158K	317.14±8.25
Cart2Pole	98K	72.94±2.81	163K	322.77±9.15	101K	78.65±9.40	101K	38.43±3.62	160K	331.09±9.91
Cart3Pole	101K	72.57±2.65	165K	320.69±4.10	104K	76.63±7.97	104K	38.77±3.98	162K	320.58±8.69
Acro3bot	99K	72.67±4.09	164K	321.44±8.06	102K	76.84±8.30	103K	38.81±3.89	160K	317.28±9.19

References

- [1] P.T. Brewick, S.F. Masri, An evaluation of data-driven identification strategies for complex nonlinear dynamic systems, *Nonlinear Dyn.* 85 (2) (2016) 1297–1318. <https://doi.org/10.1007/s11071-016-2761-x>
- [2] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, *Science* 304 (5667) (2004) 78–80. <https://doi.org/10.1126/science.1091277>
- [3] K.T. Carlberg, A. Jameson, M.J. Kochenderfer, J. Morton, L. Peng, F.D. Witherden, Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning, *J. Comput. Phys.* 395 (2019) 105–124. <https://doi.org/10.1016/j.jcp.2019.05.041>
- [4] A. Ghadami, B.I. Epureanu, Data-driven prediction in dynamical systems: recent developments, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 380 (2229) (2022) 20210213. <https://doi.org/10.1098/rsta.2021.0213>
- [5] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- [6] C. Meng, S. Griesemer, D. Cao, S. Seo, Y. Liu, When physics meets machine learning: a survey of physics-informed machine learning, *Mach. Learn. Comput. Sci. Eng.* 1 (1) (2025) 20. <https://doi.org/10.1007/s44379-025-00016-0>
- [7] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [8] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, *The Handbook of Brain Theory and Neural Networks* vol. 3361 (10) (1995) 1995.
- [9] L. Medsker, L.C. Jain, *Recurrent Neural Networks: Design and Applications*, CRC Press, 1999.
- [10] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, S. Ho, Lagrangian neural networks, 2020. [arXiv:2003.04630](https://arxiv.org/abs/2003.04630)
- [11] M. Lutter, C. Ritter, J. Peters, Deep Lagrangian networks: using physics as model prior for deep learning, in: *International Conference on Learning Representations*, 2018.
- [12] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, in: *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.
- [13] P. Toth, D.J. Rezende, A. Jaegle, S. Racanière, A. Botev, I. Higgins, Hamiltonian generative networks, in: *International Conference on Learning Representations*, 2019.
- [14] B. Lusch, J.N. Kutz, S.L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nat. Commun.* 9 (1) (2018) 4950. <https://doi.org/10.1038/s41467-018-07210-0>
- [15] Y. Xiao, X. Zhang, X. Xu, X. Liu, J. Liu, Deep neural networks with Koopman operators for modeling and control of autonomous vehicles, *IEEE Trans. Intell. Veh.* 8 (1) (2022) 135–146.
- [16] Y.D. Zhong, B. Dey, A. Chakraborty, Extending Lagrangian and Hamiltonian neural networks with differentiable contact models, in: *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 21910–21922.
- [17] M. Finzi, K.A. Wang, A.G. Wilson, Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints, in: *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 13880–13889.
- [18] A. Sosanya, S. Greydanus, Dissipative Hamiltonian neural networks: learning dissipative and conservative dynamics separately, 2022. [arXiv:2201.10085](https://arxiv.org/abs/2201.10085)
- [19] Y.D. Zhong, B. Dey, A. Chakraborty, Dissipative SymODEN: encoding Hamiltonian dynamics with dissipation and control into deep learning, 2020. [arXiv:2002.08860](https://arxiv.org/abs/2002.08860)
- [20] A. Troch, K. Mets, S. Mercelis, Action-conditioned Hamiltonian generative networks (AC-HGN) for supervised and reinforcement learning, in: *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, PMLR, 2025, pp. 310–322.
- [21] R. Grzeszczuk, D. Terzopoulos, G. Hinton, NeuroAnimator: fast neural network emulation and control of physics-based models, in: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, ACM Press, Not Known, 1998, pp. 9–20. <https://doi.org/10.1145/280814.280816>
- [22] J. Tompson, K. Schlachter, P. Sprechmann, K. Perlin, Accelerating Eulerian fluid simulation with convolutional networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 3424–3433.
- [23] M. Deisenroth, C.E. Rasmussen, PILCO: a model-based and data-efficient approach to policy search, in: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 465–472.
- [24] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.
- [25] C. Runge, Ueber die numerische Auflösung von Differentialgleichungen, *Math. Ann.* 46 (2) (1895) 167–178. <https://doi.org/10.1007/BF01446807>
- [26] W. Kutta, *Beitrag Zur Näherungsweise Integration Totaler Differentialgleichungen*, Teubner, 1901.
- [27] S. Eidnes, A.J. Stasik, C. Sterud, E. Bohn, S. Riemer-Sørensen, Pseudo-Hamiltonian neural networks with state-dependent external forces, *Phys. D Nonlinear Phenom.* 446 (2023) 133673. <https://doi.org/10.1016/j.physd.2023.133673>
- [28] M. David, F. Méhats, Symplectic learning for Hamiltonian neural networks, *J. Comput. Phys.* 494 (2023) 112495. <https://doi.org/10.1016/j.jcp.2023.112495>
- [29] Y.D. Zhong, B. Dey, A. Chakraborty, Symplectic ODE-Net: learning Hamiltonian dynamics with control, in: *International Conference on Learning Representations*, 2019.
- [30] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2022. [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
- [31] I. Higgins, P. Wirsberger, A. Jaegle, A. Botev, SyMetric: measuring the quality of learnt Hamiltonian dynamics inferred from vision, in: *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 25591–25605.
- [32] A. Ramesh, B. Ravindran, Physics-informed model-based reinforcement learning, in: *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, PMLR, 2023, pp. 26–37.
- [33] P. Jin, Z. Zhang, A. Zhu, Y. Tang, G.E. Karniadakis, SympNets: intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, *Neural Netw.* 132 (2020) 166–179. <https://doi.org/10.1016/j.neunet.2020.08.017>
- [34] Y.D. Zhong, B. Dey, A. Chakraborty, Benchmarking energy-conserving neural networks for learning dynamics from data, in: *Learning for Dynamics and Control*, PMLR, 2021, pp. 1218–1229.
- [35] N. Lambert, K. Pister, R. Calandra, Investigating compounding prediction errors in learned dynamics models, 2022. [arXiv:2203.09637](https://arxiv.org/abs/2203.09637)
- [36] P.R. Vlachas, J. Pathak, B.R. Hunt, T.P. Sapsis, M. Girvan, E. Ott, P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, 2020. [arXiv:1910.05266](https://arxiv.org/abs/1910.05266)
- [37] I. Huh, E. Yang, S.J. Hwang, J. Shin, et al., Time-reversal symmetric ODE network, *Adv. Neural Inf. Process. Syst.* 33 (2020) 19016–19027.
- [38] D.J. Rezende, F. Viola, Taming VAEs, 2018. [arXiv:1810.00597](https://arxiv.org/abs/1810.00597) [cs, stat]
- [39] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, Beta-VAE: learning basic visual concepts with a constrained variational framework, in: *International Conference on Learning Representations*, 2017.
- [40] R.M. Neal, et al., MCMC using Hamiltonian dynamics, in: *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC, 2011.
- [41] A. Botev, A. Jaegle, P. Wirsberger, D. Hennes, I. Higgins, Which priors matter? Benchmarking models for learning latent dynamics, 2021. [arXiv:2111.05458](https://arxiv.org/abs/2111.05458) [stat]