Action-Conditioned Hamiltonian Generative Networks (AC-HGN) for Supervised and Reinforcement Learning

Arne Troch ARNE.TROCH@UANTWERPEN.BE
Kevin Mets KEVIN.METS@UANTWERPEN.BE
Siegfried Mercelis SIEGFRIED.MERCELIS@UANTWERPEN.BE
University of Antwerp - imec
IDLab - Faculty of Applied Engineering
Sint-Pietersvliet 7, 2000 Antwerp, Belgium

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

This paper introduces Action-Conditioned Hamiltonian Generative Networks (AC-HGN), a physics-informed neural network architecture which learns Hamiltonian dynamics in environments subject to state-dependent external forces. AC-HGN embeds control inputs of any form into an abstract phase space, extending abstract Hamiltonian dynamics with learned external forces. In a supervised setting, results show that AC-HGN surpasses the prediction accuracy of state-of-the-art Lagrangian Neural Networks when trained on a static dataset. Furthermore, AC-HGN can be readily used as a physics-informed world model in a Model-Based Reinforcement Learning (MBRL) setting by embedding policy actions as external forces. Due to the autoencoder structure of AC-HGN, this marks the first Physics-Informed MBRL algorithm which is not reliant on any domain knowledge and is not limited to specific input modalities. Experimental results demonstrate that AC-HGN achieves competitive sample efficiency and asymptotic performance in simple environments, with minimal degradation in more complex environments, while significantly outperforming an uninformed world model. We conclude that the proposed architecture can accurately and efficiently capture environment dynamics and external forces in a Hamiltonian fashion while requiring no domain-specific knowledge, improving the applicability of physics-informed neural networks in supervised and reinforcement learning settings.

Keywords: Physics-Informed Neural Networks, Hamiltonian Neural Networks, Model-Based Reinforcement Learning

1. Introduction

Reinforcement Learning (RL) has proven to be effective in solving sequential decision-making problems through a trial-and-error optimization strategy. In modern work, this approach has been combined with high-dimensional function approximators to optimize a wide range of simulated cases across various domains such as games and robotics Haarnoja et al. (2018); Hafner et al. (2023); Schrittwieser et al. (2020). However, the extension of these simulated results to real-life cases has been shown to be non-trivial, leading to a multitude of challenges that must be addressed prior to the successful application of RL in real-world settings Dulac-Arnold et al. (2021). One of these challenges is sample efficiency, defined by Dulac-Arnold et al. as *being able to learn on live systems from limited samples*. While interacting with real-world systems is often slow, dangerous and costly, simulations are likely to be inaccurate, resulting in reduced performance when transferring trained agents from simulation to reality Zhao et al. (2020).

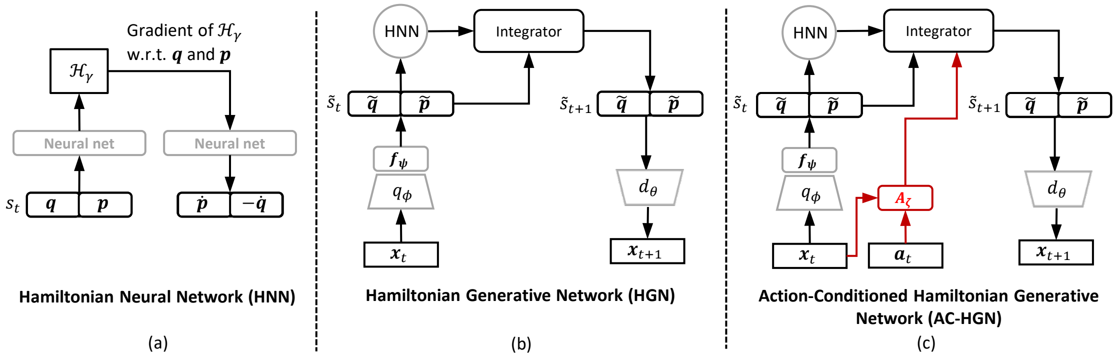

Figure 1: Hamiltonian network architectures. **HNN (a)** learns Hamiltonian dynamics through in-graph gradients. **HGN (b)** uses an autoencoder structure to learn abstract Hamiltonian dynamics from any input modality. **AC-HGN (c) (ours)** learns an abstract embedding of the external forces applied to the system.

Model-Based Reinforcement Learning (MBRL) has emerged as a compelling alternative to model-free approaches to enhance sample efficiency. By utilizing a (learned) dynamics model of the environment to perform policy updates, the required number of environment interactions can be significantly reduced. Sample efficiency (also referred to as sample complexity or data complexity) has been identified as a key advantage of MBRL in multiple studies Moerland et al. (2023); Plaat et al. (2023). Furthermore, recent MBRL approaches have demonstrated state-of-the-art results in both asymptotic performance and sample efficiency Hafner et al. (2023); Hansen et al. (2023). The success of these methods, however, is heavily dependent on the accuracy of the dynamics model. Since the policy is trained using the dynamics model, the magnitude of policy improvement is heavily reliant on the accuracy and generalization abilities of the model, especially in low-data regimes.

Recently, there has been growing interest in improving dynamics model accuracy through inductive biases. When these biases are based on physics knowledge, they create a form of physics-informed machine learning often referred to as Physics-Informed Neural Networks Karniadakis et al. (2021). Research has shown that the inclusion of physics knowledge can improve generalization performance, reaching extrapolation capabilities beyond traditional methods in data-limited environments. A promising approach explores the learning of models that inherently incorporate algebraic structures which align with the physical laws governing the system. Using these structures, these models enforce symmetries, invariances or conserved quantities serving as strong inductive biases that help guide the learning process. In particular, Hamiltonian and Lagrangian constraints have been proposed to serve this purpose Greydanus et al. (2019); Cranmer et al. (2020).

This paper proposes **Action-Conditioned Hamiltonian Generative Networks (AC-HGN)**, an extension on previous methods which allows for the incorporation of state-dependent external forces into abstract Hamiltonian dynamics learning. Using this architecture as a world model in MBRL, we combine the strengths of both approaches, further improving the sample efficiency compared to baseline model-free and model-based approaches. Our approach distinguishes itself from the current literature by requiring no domain-specific knowledge while allowing for the incorporation of external forces, significantly improving the applicability of physics-informed neural networks. Concretely, our contributions are as follows.

- We propose a physics-informed model architecture that extends abstract Hamiltonian dynamics with embedded state-dependent external forces. To our knowledge, this is the first energy-conserving neural network architecture capable of incorporating unknown state-dependent external forces into the dynamics model.
- We propose an intuitive approach to extend the learning objective of AC-HGN with multiple regularization techniques.
- We apply the proposed architecture in a model-based training loop to create a Physics-Informed MBRL algorithm which incorporates policy actions in abstract Hamiltonian dynamics.
- We perform extensive validation of our proposed architecture in both supervised and reinforcement learning settings. Supervised experiments demonstrate that AC-HGN surpasses state-of-the-art physics-informed baselines, without the need for additional regularization. When applied to MBRL, AC-HGN matches an informed baseline in terms of asymptotic performance and sample efficiency in simple environments, with minimal performance decrease in more complex environments.

2. Related Works

This paper is situated in the field of Physics-Informed Model-Based Reinforcement Learning (PIMBRL), which resides in the intersection of Physics-Informed Machine Learning and Model-Based Reinforcement Learning. In this section, we provide a brief discussion on the relevant state-of-the-art from each field of research, as well as recent advancements in PIMBRL.

Physics-Informed machine learning: Three main approaches have been proposed to embed physics knowledge in machine learning models Karniadakis et al. (2021). (1) Observational biases, where sufficient data coverage can lead the model to learn the underlying physical principles, although the required coverage is often unrealistic in practice. (2) Learning biases, which impose soft constraints through penalization of the loss function during training. (3) Inductive biases, where physics knowledge is embedded through specialized NN architectures. The most successful example of this approach is convolutional NNs LeCun and Bengio (1995), while recent trends have proposed embeddings that learn to preserve the underlying Hamiltonian Greydanus et al. (2019); Eidnes et al. (2023), Lagrangian Cranmer et al. (2020); Lutter et al. (2018), or Koopman Lusch et al. (2018) structures. These approaches have been further extended to allow complex input modalities through the use of autoencoder structures Toth et al. (2019). The architecture proposed in this work can be categorized under the idea of inductive biases, building on the concept of Hamiltonian dynamics in autoencoder architectures.

Model-Based RL: An often proposed starting point for MBRL research is the Dyna algorithm Sutton (1991). Many modern approaches build on the formulation proposed in Dyna with various extensions and improvements; using DDPG Kalweit and Boedecker (2017) or TRPO Kurutach et al. (2018) instead of Q-Learning, extending action selection with complex planning techniques Silver et al. (2017) or utilizing advanced world model architectures Hafner et al. (2019); Chua et al. (2018). MBRL relies heavily on the world model, whether for planning or environment interaction, to achieve higher sample efficiency compared to model-free counterparts. To achieve accurate world models, however, one must deal with a constantly shifting environment (caused by exploration and policy improvement) and a high risk of compounding errors when unrolling over multiple timesteps. Approaches such as Deisenroth and Rasmussen (2011); Janner et al. (2019) attempt to address such problems by accounting for uncertainty in various ways. Other approaches learn latent world models, either to improve planning speed Schrittwieser et al. (2020); Silver et al. (2018) or to handle complex image inputs Hafner et al. (2023); Hansen et al. (2023). The architecture proposed in this work follows this concept of a latent space, allowing for an abstract physics-informed world model to be learned regardless of the input modality. For a more comprehensive review of MBRL literature, we refer readers to Moerland et al. (2023); Plaat et al. (2023).

Physics-Informed Model-Based Reinforcement Learning: The combination of both techniques is novel, with a limited number of works proposing such ideas in recent literature. Liu and Wang (2021) include a learning bias into their world model, using the (partially) known governing equations from the environment to calculate residuals during training and act as a physics loss. Other approaches use the governing equations as a starting point for their dynamics model, accounting for the residual error present in these incomplete or incorrect equations with a learned black-box model Zakariae et al. (2022); Asri et al. (2024). Finally, Ramesh and Ravindran (2023) present an approach that is most similar to our work and is, to the best of our knowledge, the first PIMBRL method to incorporate an inductive bias into world model learning. They propose a separable Lagrangian world model which learns the potential energy function $\mathcal{V}(\mathbf{q})$ and a lower triangular matrix $\mathbf{L}(\mathbf{q})$, allowing the computation of the acceleration $\ddot{\mathbf{q}}$ using the mass matrix, Coriolis term, and gravitational term. Since this approach does not rely on residuals as physics knowledge, it requires significantly less knowledge about the physics of the environment. It does, however, still rely on a correct mapping between the actions taken by the RL agent and the force applied to each generalized coordinate. Furthermore, it requires the inputs to the model to be in the form of generalized coordinates $(\mathbf{q}, \dot{\mathbf{q}})$, as specified by Lagrangian mechanics.

3. Hamiltonian Mechanics

Hamiltonian mechanics describe systems using a set of generalized coordinates in a phase space $\mathbf{s} = (\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{2n}$, consisting of generalized positions $\mathbf{q} \in \mathbb{R}^n$ and generalized momenta $\mathbf{p} \in \mathbb{R}^n$, with n the number of coordinate pairs. The dynamics of the system in the space $\mathbf{s}$ are governed by the scalar Hamiltonian function $\mathcal{H}(\mathbf{q}, \mathbf{p}) : \mathbb{R}^{2n} \rightarrow \mathbb{R}^1$, which is often seen as, but not necessarily defined by, the total energy of the system. This Hamiltonian is defined such that

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

describes the time evolution of the system. Moving in the direction of the vector field defined by $\mathbf{S}_{\mathcal{H}} = (\frac{\partial \mathcal{H}}{\partial \mathbf{p}}, -\frac{\partial \mathcal{H}}{\partial \mathbf{q}})$ follows the *symplectic gradient*, keeping the output exactly constant. The formulation of 1 can be extended to include external forces, such as motor torques. Assuming external control only influences changes in generalized momenta, this generalization is defined as

$$\frac{d\mathbf{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \mathbf{g}(\mathbf{q})\mathbf{u}, \tag{2}$$

where $\mathbf{u}$ represents the external force and $\mathbf{g}(\mathbf{q})$ the input matrix which defines how the external force impacts each element of the generalized momentum vector $\mathbf{q}$, typically assumed to have full column rank Zhong et al. (2019).

Table 1: Optimal threshold values for each of the regularized AC-HGN variants, selected using a hyperparameter search. The experiment name acts an identifier of each variant.

Regularization	k_P	k_A	k_C	Experiment Name
Poisson	0.01	-	-	AC-HGN-POISS
Action Embedding	-	5.0	-	AC-HGN-ACT
Contrastive	-	-	0.00001	AC-HGN-CONTR
All	0.01	5.0	0.00001	AC-HGN-ALL

4. Learning Abstract Hamiltonians with State-Dependent External Forces

In this section we introduce an extension to the Hamiltonian Generative Network (HGN) proposed by Toth et al. (2019). HGN is itself an extension of the Hamiltonian Neural Network (HNN) Greydanus et al. (2019), which learns a Hamiltonian from data by predicting future states using the Hamiltonian dynamics described in 1. The neural network receives a set of generalized coordinates $\mathbf{s}$ as input, which it uses to output a single scalar value (i.e., the learned Hamiltonian $\mathcal{H}_\gamma$). The dynamics of the system are then predicted by taking an in-graph gradient of this scalar with respect to the input coordinates (Figure 1a). HGN extends this idea to an autoencoder structure Kingma and Welling (2022), mapping an input of any modality $\mathbf{x}_t$ into an abstract phase space $\tilde{\mathbf{s}}_t$ using a variational encoder $\mathbf{z} \sim q_\phi(\cdot|\mathbf{x}_t)$ and transformer $\tilde{\mathbf{s}}_t = f_\psi(\mathbf{z})$ (Figure 1b). This abstract phase space is then arbitrarily split into position $\tilde{\mathbf{q}}$ and momentum $\tilde{\mathbf{p}}$ to be used as input to a HNN $\mathcal{H}_\gamma$ which, together with an integrator, acts as the latent dynamics of the network $\tilde{\mathbf{s}}_{t+1} = Integrator(\mathcal{H}_\gamma(\tilde{\mathbf{s}}_t))$. A decoder is used to map the abstract state space to the input space $\hat{\mathbf{x}}_t = d_\theta(\tilde{\mathbf{q}}_t)$.

4.1. Action-Conditioned Hamiltonian Generative Network

The HGN proposed by Toth et al. (2019) assumes an energy-conserving system, using properties of their latent Hamiltonian dynamics to conserve energy when unrolling over time. However, this approach is not applicable to control settings, where external forces are likely to have a significant impact on the dynamics of a system. Therefore, we propose an extension to HGN which follows the dynamics proposed in 2. This approach introduces an action embedding network $\mathbf{g}(\mathbf{q})\mathbf{u} = A_\zeta(\mathbf{x}_t, \mathbf{a}_t)$ which, separate from the embedding of the coordinates, learns to embed the input $\mathbf{x}_t$ and control action $\mathbf{a}_t$ into an abstract external force (Figure 1c). This is similar to the approach proposed by Eidnes et al. (2023), however, their embedding is generated based on generalized coordinates as input whereas we generate an abstract embedding from inputs of any modality. Based on 2, the abstract of dynamics of the system are adjusted to incorporate the action embedding:

$$\tilde{\mathbf{s}}_{t+1} = (\tilde{\mathbf{q}}_{t+1}, \tilde{\mathbf{p}}_{t+1}) = \left(\tilde{\mathbf{q}}_t + \frac{\partial \mathcal{H}_\gamma}{\partial \tilde{\mathbf{p}}_t} dt, \tilde{\mathbf{p}}_t - \frac{\partial \mathcal{H}_\gamma}{\partial \tilde{\mathbf{q}}_t} dt + A_\zeta(\mathbf{x}_t, \mathbf{a}_t)\right). \tag{3}$$

The overall optimization objective of the **Action-Conditioned HGN (AC-HGN)** is given by:

$$\mathcal{L}(\phi, \psi, \gamma, \theta, \zeta; \mathbf{x}_0, \ldots, \mathbf{x}_T) = \frac{1}{T+1} \sum_{t=0}^{T} \left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)} \left[\log d_\theta(\hat{\mathbf{x}}_t|\tilde{\mathbf{s}}_t)\right]\right] - \mathrm{KL}\left[q_\phi(\mathbf{z})||p(\mathbf{z})\right], \tag{4}$$

given a sequence of $T+1$ observations and $p(\mathbf{z}) = \mathcal{N}(0, \mathbb{I})$ a unit Gaussian prior. We condition the decoder on $\tilde{\mathbf{s}}_t$ rather than $\tilde{\mathbf{q}}_t$ as in the original work since observations $(\mathbf{x}_t)$ in control settings often include derivatives, which are dependent on both position and momentum.

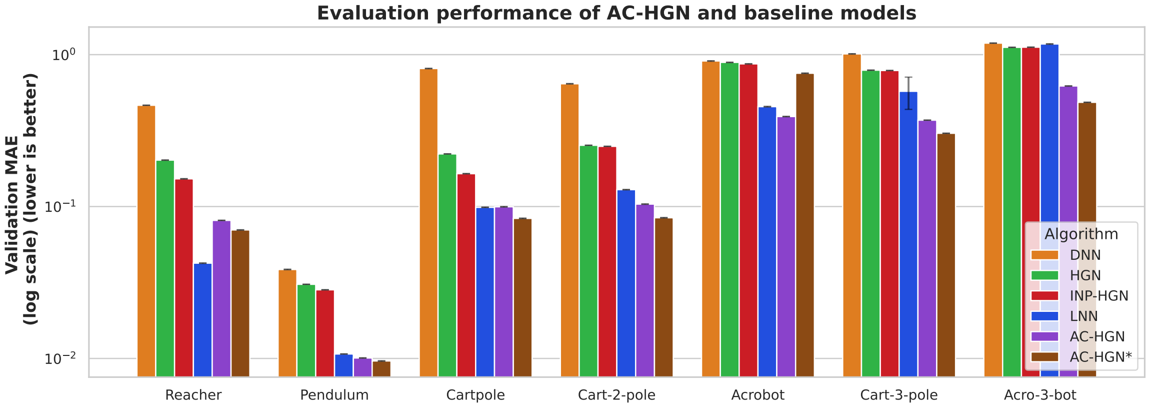

Figure 2: Average test error across 10000 unseen trajectories of length 32 for AC-HGN (ours), AC-HGN* (ours) and baselines. The errorbars indicate 95% confidence intervals.

4.2. Regularization in AC-HGN

Similar to HGN Toth et al. (2019), the learning objective of AC-HGN is defined following the GECO algorithm proposed in Rezende and Viola (2018). GECO introduces an intuitive approach to tune the loss, leading to the following optimization objective:

$$\mathcal{L}(\phi,\psi,\gamma,\theta,\zeta;\mathbf{x}_0,\ldots,\mathbf{x}_T) = \text{KL}\left[q_\phi(\mathbf{z})||p(\mathbf{z})\right] + \lambda\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)}\left[\mathcal{C}_k(\mathbf{x}_t, d_\theta(\hat{\mathbf{x}}_t|\tilde{\mathbf{s}}_t))\right], \tag{5}$$

where $\mathcal{C}_k = [||(d_\theta(\hat{\mathbf{x}}_t|\tilde{\mathbf{s}}_t) - \mathbf{x}_t)||_2 - k]$ functions as a reconstruction constraint, with threshold k and λ updated each timestep following $\lambda^t \leftarrow \lambda^{t-1}\text{exp}(\propto \mathcal{C}_k^t)$ to ensure positivity. Previous works have proposed regularization of Hamiltonian Networks to further direct the model towards realistic solutions Greydanus et al. (2019); Eidnes et al. (2023). We investigate the effect of such regularizations on the AC-HGN by extending the optimization objective proposed in 5:

$$\begin{aligned}\mathcal{L}(\ldots) =& \text{KL}\left[q_\phi(\mathbf{z})||p(\mathbf{z})\right] \\ &+ \lambda_R\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)}\left[\mathcal{C}_{k_R}(\mathbf{x}_t, d_\theta(\hat{\mathbf{x}}_t|\tilde{\mathbf{s}}_t))\right] && \text{(Reconstruction)} \\ &+ \lambda_P\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)}\left[\mathcal{C}_{k_P}((\tilde{\mathbf{q}}_t, \tilde{\mathbf{q}}_{t+1}); \tilde{\mathbf{p}}_t)\right] && \text{(Poisson)} \\ &+ \lambda_A\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)}\left[\mathcal{C}_{k_A}(A_\zeta)\right] && \text{(Action Regularization)} \\ &+ \lambda_C\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_1,\ldots,\mathbf{x}_T)}\left[\mathcal{C}_{k_C}(\mathcal{H}_\gamma(\tilde{\mathbf{s}}_{t-1}), f_\psi(q_\phi(\mathbf{z}|\mathbf{x}_t)\right] && \text{(Contrastive)},\end{aligned} \tag{6}$$

where $\mathcal{C}_{k_R}$ the reconstruction constraint from 5, $\mathcal{C}_{k_P} = [||(\mathbf{p}_t - (\mathbf{q}_t - \mathbf{q}_{t+1}))||_2 - k_R]$ a Poisson bracket constraint, $C_{k_A} = [||\zeta||_2 - k_A]$ an L^2-norm constraint on the action embedding network parameters, and $\mathcal{C}_{k_C} = [||(f_\psi(q_\phi(\mathbf{z}|\mathbf{x})) - \mathcal{H}_\gamma(\tilde{\mathbf{s}}_{t-1}))||_2 - k_C]$ a contrastive constraint.

5. Results

We investigate the performance of AC-HGN in two learning paradigms: **supervised learning** and **reinforcement learning**. We propose two versions of our network, one which uses the delta time between two timesteps as its only domain-specific knowledge (AC-HGN) and the other which learns

Table 2: Test MAE of AC-HGN (ours), AC-HGN* (ours) and baselines.

Environment	DNN	HGN	INP-HGN	LNN	AC-HGN	AC-HGN*
Reacher	0.464	0.202	0.152	**0.042**	0.081	0.070
Pendulum	0.039	0.031	0.028	**0.011**	**0.010**	**0.010**
Cartpole	0.810	0.222	0.165	0.099	0.100	**0.084**
Cart-2-pole	0.643	0.253	0.249	0.129	0.104	**0.084**
Acrobot	0.908	0.889	0.869	0.455	**0.391**	0.754
Cart-3-pole	1.011	0.790	0.786	0.574	0.370	**0.304**
Acro-3-bot	1.190	1.117	1.119	1.175	0.622	**0.485**
Overall	0.724	0.501	0.481	0.355	**0.240**	**0.256**

this delta time as an internal parameter, therefore requiring no external information (AC-HGN*). We use a fully-connected version of the HGN architecture since we are working with state-based inputs. Our network trains by unrolling for 16 timesteps using backpropagation through time. In contrast to Toth et al. (2019), we use Euler updates since the more computationally expensive Leapfrog integrator showed no significant performance improvements. We use the supervised learning setting to investigate the impact of the regularizations proposed in 4.2. For each regularization, the optimal threshold value k, shown in Table 1, was identified using a hyperparameter sweep. We compare our results with the LNN and DNN approaches proposed in Ramesh and Ravindran (2023), which we consider the state-of-the-art which most closely matches our work. The LNN represents a physics-informed baseline, though it relies on domain-specific knowledge about the external forces and requires generalized coordinates as input, while the DNN functions as an uninformed baseline. To allow for this comparison, we perform experiments on the seven control environments proposed in their work.

5.1. Supervised Learning

For each environment, we train on 7200 trajectories of length 1000 using randomly selected control inputs. All results shown are calculated as the Mean Absolute Error (MAE) on a separate test dataset consisting of 10000 trajectories of length 32. By testing predictions across longer trajectories, we can investigate generalization towards longer, unseen time periods.

AC-HGN compared with baselines: In addition to the LNN and DNN baselines, we also compare AC-HGN with a vanilla HGN and a HGN which is given the chosen control values as input (INP-HGN). Figure 2 shows the test MAE error on each environment for the baseline architectures as well as AC-HGN and AC-HGN*, with the exact MAE values shown in Table 2. The results indicate that, across environments, the DNN architecture is unable to accurately predict environment dynamics, while the LNN and AC-HGN architectures perform best. We further observe that, in dynamically complex environments (Acrobot, Cart-2-Pole, Cart-3-Pole), AC-HGN and AC-HGN* achieve significantly lower prediction errors compared to LNN, with minor differences in most simple environments. The vanilla HGN network performs significantly worse compared to our AC-HGN approach due to its inability to account for external forces. The naive approach of INP-HGN, which is given all the required information to model the environment dynamics, does not lead to a statistically relevant impact on performance compared to HGN. Since INP-HGN and AC-HGN have access to the exact same information, this difference in performance shows that the improvements gained by AC-HGN can be contributed to the separate action embedding proposed in 4.1, rather than the inclusion of the external forces in the input space. Finally, the results also show that AC-HGN and AC-HGN* achieve similar performance, indicating that the AC-HGN architecture is able to accurately predict environment dynamics without any external knowledge required.

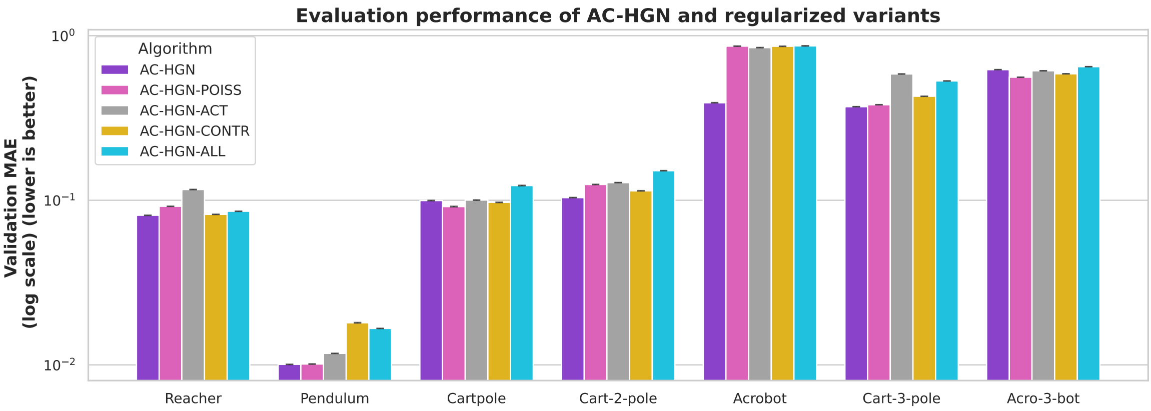

Figure 3: Average test error across 10000 unseen trajectories of length 32 for AC-HGN and its regularized variants. The errorbars indicate 95% confidence intervals.

Table 3: Test MAE of AC-HGN and its regularized variants per environment.

Environment	AC-HGN	AC-HGN-POISS	AC-HGN-ACT	AC-HGN-CONTR	AC-HGN-ALL
Reacher	**0.081**	0.092	0.116	**0.082**	0.086
Pendulum	**0.010**	**0.010**	**0.012**	0.018	0.017
Cartpole	0.100	**0.092**	0.100	0.097	0.123
Cart-2-pole	**0.104**	0.125	0.128	0.114	0.151
Acrobot	**0.391**	0.864	0.846	0.862	0.867
Cart-3-pole	**0.370**	**0.381**	0.585	0.429	0.531
Acro-3-bot	0.622	**0.559**	0.612	0.587	0.648
Overall	**0.240**	0.303	0.343	0.313	0.346

Regularized variants of AC-HGN: We investigate the impact of regularization on the behavior of AC-HGN in Figure 3 and Table 3. Contrary to the conclusions from Greydanus et al. (2019) regarding Poisson and contrastive constraints and Eidnes et al. (2023) regarding action embedding constraints, none of the proposed approaches lead to improved overall performance. Furthermore, combining all regularization approaches leads to worse overall performance. These results, combined with the strong prediction performance of AC-HGN, indicate that the standard learning objective of AC-HGN already causes the model to learn physically viable dynamics without requiring the need for additional regularization. Furthermore, the reduced average performance of the regularized objectives suggest that the added complexity in the learning objective can hinder the learning process, leading to suboptimal solutions. Nevertheless, we believe that further research is required to conclusively determine the usability of such regularizations in HNN architectures.

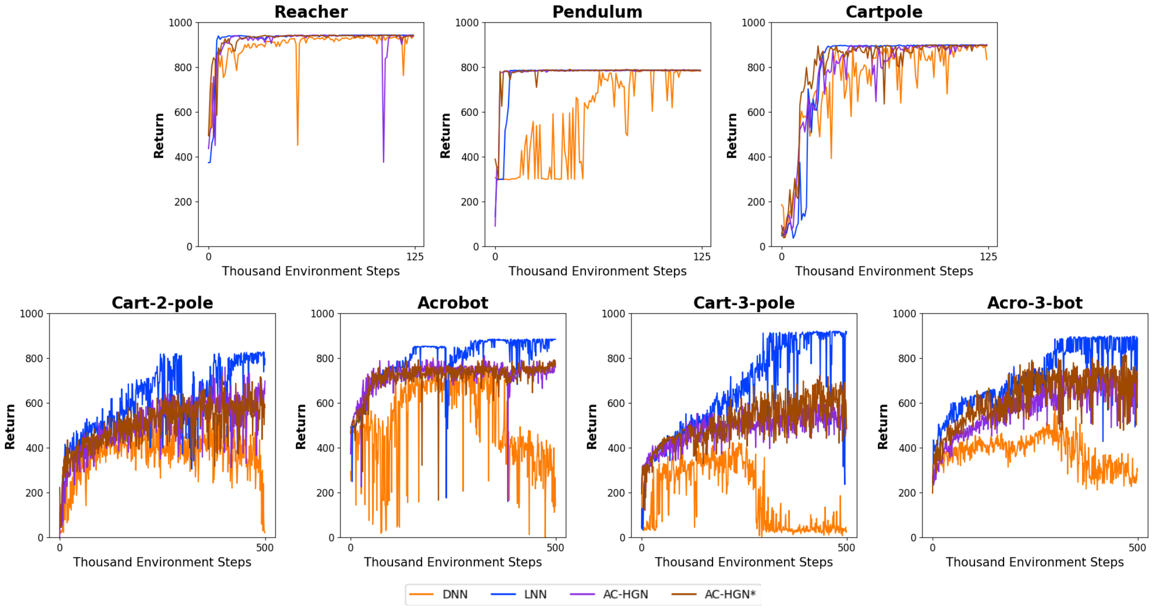

Figure 4: RL training curves for AC-HGN (ours), AC-HGN* (ours) and baselines. The y-axis shows the episodic return, the x-axis represents the number of environment steps.

5.2. Reinforcement Learning

We use AC-HGN and AC-HGN* as a physics-informed world model in the MBRL algorithm proposed by Ramesh and Ravindran (2023) which combines the model learning of LNN with a behavior learning approach inspired by the Dreamer algorithm Hafner et al. (2019, 2023). By substituting their LNN world model with our AC-HGN architecture, we create a Physics-Informed MBRL approach which has no reliance on domain-specific knowledge (in the case of AC-HGN*) through its use of abstract Hamiltonian dynamics. We compare the results of AC-HGN and AC-HGN* with the informed LNN and uniformed DNN baselines using multiple evaluation metrics. To investigate the training process and final return, we present the training curves in Figure 4. These curves show that the AC-HGN approaches perform significantly better than the uninformed DNN baseline. Compared to the informed LNN architecture, AC-HGN and AC-HGN* match the training curves of the LNN world model in simple environments, with decreasing performance when the environment complexity increases. While these results seem to contradict the conclusions from the supervised experiments, the supervised experiments do not take into account the speed of learning of the proposed architectures. Therefore, a more detailed analysis of sample efficiency is required.

To achieve this, we use the **global normalized regret** metric, proposed in Dulac-Arnold et al. (2021) as an indication of sample efficiency during the training process. Global normalized regret represents the return that was lost prior to convergence due to poor policy performance. The results, shown in Table 4, indicate that AC-HGN, and especially AC-HGN* manage to be more sample efficient than LNN in simple environments (Reacher, Pendulum, Cartpole), while LNN achieves superior sample efficiency in the complex environments. Overall, the results also indicate similar or

Table 4: Global Normalized Regret of AC-HGN (ours), AC-HGN* (ours) and baselines.

Environment	DNN	LNN	AC-HGN	AC-HGN*
Reacher	5.913	2.820	3.144	**2.358**
Pendulum	40.763	18.142	**16.010**	**15.928**
Cartpole	23.959	19.956	19.892	**16.692**
Cart-2-pole	276.371	**171.328**	219.137	217.859
Acrobot	199.081	**75.427**	108.762	112.792
Cart-3-pole	366.380	**146.811**	226.024	215.094
Acro-3-bot	271.992	**117.184**	183.523	158.290
Overall	169.208	**78.810**	110.927	105.573

better performance for AC-HGN* compared to AC-HGN, again indicating that our approach is not dependent on this external information.

6. Conclusion and Future Work

This paper introduces an extension on abstract Hamiltonian dynamics learning which embeds external forces into the latent phase space. Our proposed architecture, named Action-Conditioned Hamiltonian Generative Networks (AC-HGN), learns to embed actions taken by an agent into forces applied to abstract phase-space dynamics. We extend the learning objective of AC-HGN to intuitively incorporate multiple regularization approaches. Trough experimental validation in a supervised setting, we demonstrate that our approach outperforms state-of-the-art physics-informed baselines, without requiring any domain-specific knowledge about the environment. Our experiments also show that regularized training objectives do not lead to an overall increase in performance, suggesting that the standard learning objective of AC-HGN is sufficient to learn physically realistic dynamics. We apply our world model in a Model-Based Reinforcement Learning algorithm, combining the strengths of both fields to further improve sample efficiency compared to model-free and uninformed model-based approaches. Experiments in this setting show that our AC-HGN world model significantly outperforms an uformed baseline in both sample efficiency and asymptotic performance. Compared to an informed Lagrangian Neural Network baseline, AC-HGN achieves competitive results in simple environments with limited loss of performance in more dynamically complex problems.

This work introduces multiple avenues for future work. First, the ideas proposed in this work can be readily applied to Lagrangian Neural Networks which, due to their more natural inclusion of external forces, form an interesting candidate for abstract latent dynamics in a reinforcement learning setting. Second, while the experiments conducted in this paper indicate that regularization does not improve performance, further validation of these ideas using other regularization techniques, other network architectures and more environments seems necessary to draw more definitive conclusions. Third, we believe that many modern advancements in MBRL, such as ensemble networks to combat uncertainty Janner et al. (2019) and pessimistic world models to handle robustness Herremans et al. (2024), could be readily applied to the physics-informed world model proposed in this paper.

Acknowledgments

This work was supported by the Research Foundation Flanders (FWO) under Grant Number 1SHAH24N.

References

Zakariae EL Asri, Olivier Sigaud, and Nicolas Thome. Physics-Informed Model and Hybrid Planning for Efficient Dyna-Style Reinforcement Learning. In *Reinforcement Learning Conference*, November 2024.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian Neural Networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, February 2020.

Marc Deisenroth and Carl 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)*, pages 465–472, 2011.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, September 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05961-4.

Sølve Eidnes, Alexander J. Stasik, Camilla Sterud, Eivind Bøhn, and Signe Riemer-Sørensen. Pseudo-Hamiltonian neural networks with state-dependent external forces. *Physica D: Nonlinear Phenomena*, 446:133673, April 2023. ISSN 0167-2789. doi: 10.1016/j.physd.2023.133673.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1861–1870. PMLR, July 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2555–2565. PMLR, May 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models, January 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *The Twelfth International Conference on Learning Representations*, October 2023.

Siemen Herremans, Ali Anwar, and Siegfried Mercelis. Robust Model-Based Reinforcement Learning with an Adversarial Auxiliary Model. In *First Reinforcement Learning Safety Workshop*, August 2024.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Gabriel Kalweit and Joschka Boedecker. Uncertainty-driven Imagination for Continuous Deep Reinforcement Learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 195–206. PMLR, October 2017.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021. doi: 10.1038/s42254-021-00314-5.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Xin-Yang Liu and Jian-Xun Wang. Physics-informed Dyna-style model-based deep reinforcement learning for dynamic control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2255):20210618, November 2021. doi: 10.1098/rspa.2021.0618.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

Michael Lutter, Christian Ritter, and Jan Peters. Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In *International Conference on Learning Representations*, September 2018.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based Reinforcement Learning: A Survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, January 2023. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000086.

Aske Plaat, Walter Kosters, and Mike Preuss. High-accuracy model-based reinforcement learning, a survey. *Artificial Intelligence Review*, 56(9):9541–9573, September 2023. ISSN 1573-7462. doi: 10.1007/s10462-022-10335-w.

Adithya Ramesh and Balaraman Ravindran. Physics-Informed Model-Based Reinforcement Learning. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, pages 26–37. PMLR, June 2023.

Danilo Jimenez Rezende and Fabio Viola. Taming VAEs, October 2018.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, December 2020. ISSN 1476-4687. doi: 10/ghqh6d.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 1476-4687. doi: 10.1038/nature24270.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aar6404.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377.

Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian Generative Networks. In *International Conference on Learning Representations*, September 2019.

E. L. Zakariae, Clément Rambour, L. E. Vincent, and Nicolas THOME. Residual Model-Based Reinforcement Learning for Physical Dynamics. In *3rd Offline RL Workshop: Offline RL as a"Launchpad"*, 2022.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, December 2020. doi: 10.1109/SSCI47803.2020.9308468.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control. In *International Conference on Learning Representations*, September 2019.