Hamiltonian GAN

Christine Allen-Blanchette CA15@PRINCETON.EDU
Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08540

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

A growing body of work leverages the Hamiltonian formalism as an inductive bias for physically plausible neural network based video generation. The structure of the Hamiltonian ensures conservation of a learned quantity (e.g., energy) and imposes a phase-space interpretation on the low-dimensional manifold underlying the input video. While this interpretation has the potential to facilitate the integration of learned representations in downstream tasks, existing methods are limited in their applicability as they require a structural prior for the configuration space at design time. In this work, we present a GAN-based video generation pipeline with a learned configuration space map and Hamiltonian neural network motion model, which allow us to learn a representation of the configuration space from data. We train our model with a physics-inspired cyclic-coordinate loss function which encourages a minimal representation of the configuration space and improves interpretability. We demonstrate the efficacy and advantages of our approach on the Hamiltonian Dynamics Suite Toy Physics dataset.

Keywords: Dynamics Learning, Structure-Preserving Neural Networks, Physics-Informed Machine Learning, Generative modeling

1. Introduction

Hamiltonian mechanics can be applied to systems on general manifolds, and is independent of a particular choice of coordinates (Holm et al. (2009)). This flexibility is attractive in data-driven modelling learning where low-dimensional manifolds are often approximated from high-dimensional data (Bengio et al. (2013)). In this work, we leverage the structure of the Hamiltonian within the generative adversarial network (GAN) (Goodfellow et al. (2014)) framework for physics-guided video generation. The problem of video generation asks for a representation of video frames and frame-to-frame transitions. The Hamiltonian formalism provides a useful framing for this problem: each video frame is an observation of the system state; the set of all states is the phase-space; and state transitions are determined by Hamilton's equations. With this framing, the time evolution of the video-generating process is determined by well-understood physical principles which gives an interpretation to the learned representation which can be useful for downstream tasks such as control.

The Hamiltonian formalism has been used as an inductive bias in physics-guided video generation in several recent works (Toth et al. (2019); Saemundsson et al. (2020); Zhong and Leonard (2020); Higgins et al. (2021)). In each of the aforementioned, a variational autoencoder (VAE) (Kingma and Welling (2013)) is used as the generative model and a Hamiltonian neural network (HNN) (Greydanus et al. (2019)) (or symplectic recurrent neural network (SRNN) (Chen et al. (2019))) is used to compute trajectories in the latent space. The use of the HNN model allows for an interpretation of the latent space as a phase-space for the dynamical system underlying the video. While this interpretation has the potential to facilitate the integration of learned representations in downstream tasks such as control, a persistent challenge has been in the identification of an appropriate structural prior for the phase-space.

The authors in Toth et al. (2019); Saemundsson et al. (2020) and Higgins et al. (2021) circumvent this challenge by using a standard Gaussian prior and a phase-space of substantially higher dimension than the dynamical system underlying the video. While this approach is common in the VAE literature, it can lead to solutions with poor reconstructive ability; a limitation discussed in Davidson et al. (2018) and Dai and Wipf (2019), and observed in Zhong and Leonard (2020) and Botev et al. (2021); which can limit the utility of learned representations in downstream tasks. In contrast, the authors in Zhong and Leonard (2020) use prior knowledge of the configuration space to select a distribution with the appropriate structure. While this lends greater interpretability to the latent space, it requires prior knowledge of the structure of the phase-space and the availability of a distribution on that space that admits the reparameterization trick (Kingma and Welling (2013)).

In our work, we use a generative adversarial network (GAN) (Goodfellow et al. (2014)) based video generation approach to circumvent the challenge of identifying an appropriate distribution in advance by implicitly learning the structure of the phase-space from data. We take inspiration from other GAN-based video generation pipelines (e.g., Tulyakov et al. (2018); Yoon et al. (2019); Gordon and Parde (2021); Skorokhodov et al. (2021)) where motion and content vectors are initialized with Gaussian random noise, propagated forward in time using a recurrent neural network, and mapped into the image space by a generator network. To allow for the interpretation of the motion vectors as elements of phase-space for the underlying dynamical system, we incorporate three key elements: (1) a configuration-space map that learns to transform Gaussian random vectors to an intermediate space which we interpret as the configuration space of the system; (2) a HNN module in place of the standard discrete recurrent unit to enforce continuous and conservative latent dynamics, and (3) a physics-inspired cyclic-coordinate loss function that encourages a sparsity in the configuration space representation allowing for the discovery of phase-space structure.

We empirically evaluate our model on the Hamiltonian Dynamics Suite Toy Physics dataset (Toth et al. (2019)). Our analysis highlights the importance of the HNN module for temporal coherence of generated sequences, and of the configuration-space map and cyclic-coordinate loss for a compact and interpretable representation of the motion subspace.

2. Related work

In the physics-guided learning literature, a growing number of works investigate the integration of Hamiltonian and Lagrangian formalisms as an inductive bias for dynamical systems forecasting. The authors in Greydanus et al. (2019) learn to predict the change in position and momentum for conservative mechanical systems using a system specific Hamiltonian parameterized by a neural network. The authors in Chen et al. (2019) find that using the Hamiltonian neural network as a recurrent module improves long term prediction accuracy without the need for gradient information. This observation is leveraged in Zhong et al. (2019); Lutter et al. (2019); Roehrl et al. (2020) for control, Allen-Blanchette et al. (2020); Zhong and Leonard (2020) for video prediction and Toth et al. (2019); Saemundsson et al. (2020) for variational autoencoding (VAE) (Kingma and Welling (2013)) based video generation. To the best of our knowledge, we are the first to use the Hamiltonian or Lagrangian formalism as an inductive bias in generative adversarial network (GAN) (Goodfellow et al. (2014)) based video generation.

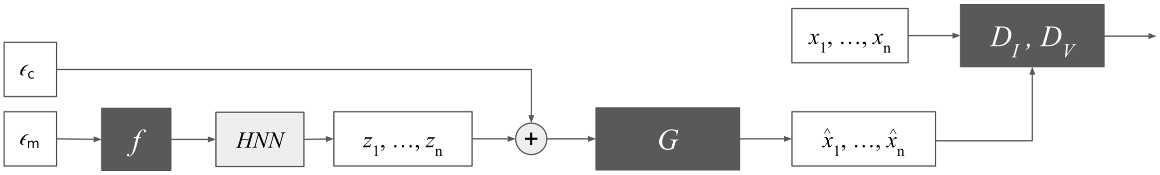

Figure 1: HGAN video generation pipeline. Physically plausible videos of conservative systems are generated from motion and content Gaussian random samples ϵ_m and ϵ_c. The motion sample ϵ_m is mapped to an initial condition z_m^0 of the underlying dynamical system by the configuration space map f, then integrated forward in time using Hamilton's equations and the learned Hamiltonian $\mathcal{H}_\theta$ in the HNN module. Each element of the resulting sequence z_m^t is concatenated with the content sample ϵ_c, then passed to the generator network G_I to produce an image $\hat{x}^t$. During training, individual images are passed to the discriminator D_I and image sequences $\{\hat{x}^j, \ldots, \hat{x}^{j+T}\}$ are passed to the discriminator D_V.

In contrast to VAE-based video generation methods which impose an explicit prior, GAN-based video generation methods implicitly learn the data distribution during training. The authors in Vondrick et al. (2016) learn to map Gaussian random noise vectors to image sequences using a generator network with 3D deconvolutional layers. The authors in Tulyakov et al. (2018); Yoon et al. (2019) find they are able to substantially reduce complexity and improve performance by decomposing the Gaussian random noise vector into motion and content vectors, propagating them forward in time using a recurrent neural network, then mapping each resulting vector to an image using a generator network with 2D deconvolutional layers. This high-level structure is used in many recent GAN-based video generation methods. Some of these methods introduce alternatives to the standard convolutional generator network – the authors in Yu et al. (2022) use alternative architecture types, the authors in Tian et al. (2021) use large pre-trained image generator networks, and the authors in Wang et al. (2020) use multiple generator networks. Others have worked to increase the expressivity of the latent code – the authors in Wang et al. (2020) use a content-motion integration scheme to improve latent space disentanglement, and the authors in Skorokhodov et al. (2021) learn a map from the standard Gaussian distribution to an intermediate distribution which may better represent the data distribution. Still others have introduced alternative motion models – the authors in Tian et al. (2021) use an RNN-based motion model to learn dynamical updates rather than subsequent states, and the authors in Gordon and Parde (2021) replace the discrete recurrent unit with the continuous NeuralODE (Chen et al. (2018)).

The work in Toth et al. (2019) and Gordon and Parde (2021) are most similar to ours. Toth et al. (2019) use the Hamiltonian formalism as an inductive bias for VAE-based video generation. Their framework generates physically plausible video sequences at different time scales and time directions. We also use a Hamiltonian neural network for video generation and thereby inherit these capabilities; by using the GAN framework, however, we circumvent the challenge of selecting an appropriate prior distribution in advance of training. Gordon and Parde (2021) use the Neural ODE (Chen et al. (2018)) recurrent unit to learn continuous dynamics in a GAN-based video generation pipeline. They demonstrate performance commensurate with other GAN based video generation approaches while benefiting from a continuous motion model. Our motion model is distinct from Gordon and Parde (2021), however, since we enforce continuous and conservative dynamics on a transformed latent space which we interpret as the underlying configuration space. Moreover, we impose a novel cyclic-coordinate loss function to encourage sparsity in the configuration space representation to facilitate discovery of phase-space structure.

3. Background

In this section we present Hamilton's equations and describe how they can be used to identify cyclic coordinates. We also present Hamiltonian neural networks (Greydanus et al. (2019)), and generative adversarial neural networks (Goodfellow et al. (2014)) as themes from each of these appear in the proposed approach.

3.1. Hamilton's equations and cyclic coordinates

The Hamiltonian formalism is a reformulation of Newton's second law, $F = ma$, in terms of the system energy rather than its forces (Goldstein et al. (2002)). This formulation is preferable in settings such as ours where the identification and quantification of system forces may be challenging. The Hamiltonian of a system, $\mathcal{H} : T^*\mathcal{Q} \mapsto \mathbb{R}$, is a map from generalized position and momentum coordinates $(q, p) \in T^*\mathcal{Q}$, to the total energy of the system, where $T^*\mathcal{Q}$ denotes the co-tangent space of the system configuration space $\mathcal{Q}$. The behavior of a Hamiltonian system is governed by Hamilton's equations:

$$\dot{p} = -\frac{\partial E}{\partial q}, \; \dot{q} = \frac{\partial E}{\partial p}, \quad E = \mathcal{H}(q, p). \tag{1}$$

Given the system Hamiltonian and its initial conditions, we can forecast future system states by integrating Hamilton's equations forward in time.

We can also use Hamilton's equations to identify cyclic coordinates. The generalized position coordinate q_i is called cyclic or ignorable if it does not contribute to the total energy of the system:

$$-\frac{\partial E}{\partial q_i} = 0 = \dot{p}_i. \tag{2}$$

From equation 2, we see that for a cyclic coordinate q_i, the corresponding conjugate momentum p_i is conserved (Goldstein et al. (2002)).

Previous work using Hamilton's equations for dynamical systems forecasting either assume prior knowledge of the dimension of the configuration space for improved interpretability (Allen-Blanchette et al. (2020); Zhong and Leonard (2020)), or select the dimension to be arbitrarily large for model flexibility (Toth et al. (2019); Saemundsson et al. (2020)). In our approach we achieve interpretability with an arbitrarily large latent dimension by incorporating our novel cyclic-coordinate loss function. Our loss function, detailed in section 4.2, encourages the identification of cyclic coordinates by regularizing the change in momentum.

3.2. Hamiltonian Neural Networks

Hamiltonian neural networks (HNNs) (Greydanus et al. (2019); Chen et al. (2019)) model the underlying dynamics of sequential data using the Hamiltonian formalism. The data are assumed to be sampled from a continuous and conservative dynamical system, and the system Hamiltonian is assumed to be unknown.

HNNs are one of a larger class of models termed NeuralODEs (Chen et al. (2018)), that use neural networks to parameterize the ordinary differential equations (ODEs) governing the dynamics underlying sequential data. NeuralODEs learn a neural network f_θ, with parameters θ, to predict a next state, $x(t_1)$, from a current state $x(t_0)$ by the following:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f_\theta(x(t_0), t)\, dt, \qquad \frac{dx(t)}{dt} = f_\theta(x(t), t).$$

In HNNs, the system Hamiltonian is parameterized using a neural network and next state predictions are formed by integrating Hamilton's equations (equation 1).

HNNs have been used for dynamical systems forecasting when position-momentum values are assumed to be known (Greydanus et al. (2019); Chen et al. (2019)) and in VAE-based video generation pipelines where ground truth position-momentum values are assumed to be unknown (Toth et al. (2019); Saemundsson et al. (2020)). As far as we know, we are the first to use HNNs for GAN-based video generation.

3.3. Generative Adversarial Networks (GANs)

In the generative adversarial network (GAN) (Goodfellow et al. (2014)) framework two networks – the generator network G, and discriminator network D – are learned with opposing objectives. The objective of the generator is to learn to map samples from a known distribution $p_{\text{z}}(z)$, to a distribution that resembles the training data distribution $p_{\text{data}}(x)$. The objective of the discriminator is to discern whether or not a given sample is from the training set and indicate this with a 1 or 0 respectively. The formulation of these contrasting objectives proposed in (Goodfellow et al. (2014)), is the following:

$$\begin{aligned} \min_G \max_D \mathcal{L}(D, G) = \mathbb{E}_{z \sim p_{\text{z}}(z)} \log[1 - D(G(z))] \\ + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)]. \end{aligned}$$

The objective is maximized when the discriminator correctly labels synthetic samples, $G(z)$, with zero and samples from the training dataset, x, with one. The objective is minimized when the generator produces synthetic samples the discriminator incorrectly labels with one.

4. Hamiltonian GAN

In this section we detail our model, a three-stage GAN-based video generation pipeline in which a configuration-space map and cyclic-coordinate loss function encourage identification of a minimal configuration-space; and a HNN motion model supports generation of physically plausible video.

4.1. Video generation

Our inference framework consists of a configuration-space map $f : \mathbb{R}^n \mapsto \mathbb{R}^k$, an HNN (Chen et al. (2019)) motion model with a learned Hamiltonian $\mathcal{H} : T^*\mathcal{Q} \mapsto \mathbb{R}$, and an image generator $G_I : \mathcal{Z} \mapsto \mathcal{I}$. We begin by sampling a Gaussian random noise vector z and decomposing it into a motion vector z_m and content vector z_c. We assume the structure of the motion subspace can be learned by our piecewise continuous configuration map f, and that the underlying dynamical system is continuous and conservative.

Given the Gaussian sample z_m, the configuration-space map f, a two layer MLP, outputs a vector y_0 which is interpreted as an initial condition on the phase-space $T^*\mathcal{Q}$. This initial condition is passed to the HNN motion model which computes future system states $y_j = (q_j, p_j)$ using Hamilton's equations (equation 1) and the learned Hamiltonian $\mathcal{H}$. Concretely, the sequence of motion vectors $\{y_j\}_{j=0}^{N-1}$, where N is the sequence length, is given by:

$$y_0 = f(z_m); \quad y_j = y_{j-1} + \int_{t_{j-1}}^{t_j} \dot{y}_{j-1}\, dt,\ j > 0 \tag{3}$$

where $\dot{y}_j = (\dot{q}_j, \dot{p}_j)$ is given by Hamilton's equations (equation 1). In our implementation, we compute the integral (equation 3) using leapfrog integration with $dt = 0.05$. While a simpler integration scheme can be used, leapfrog integration preserves simplicity and has been shown to outperform other integration techniques (Hairer et al. (2006)).

Given the content vector z_c and motion sequence $\{y_j\}_{j=0}^{N-1}$, we form the latent sequence $\{w_j\}_{j=0}^{N-1}$ where $w_j = (z_c,\, y_j)$. Given this latent sequence, we generate a synthetic video $\tilde{v} = \{\tilde{x}_j\}_{j=0}^{N-1}$ where images $\tilde{x}_j = G_I(w_j)$, are determined by the generator network.

4.2. Cyclic-coordinate loss

Our configuration-space map f defines an intermediate space between the sampling distribution, and the motion model determined by the learned Hamiltonian $\mathcal{H}$. We interpret this intermediate space as a configuration space for the dynamical system underlying the video data. Since its dimension is selected arbitrarily at design time, without constraint, the learned representation may be dispersed with little identifiable structure. To encourage a minimal representation and improve interpretability, we introduce a cyclic-coordinate loss function inspired by the observation that the momentum coordinate p_i, corresponding to a cyclic-coordinate q_i is conserved (see Section 3.1). We define our loss:

$$\mathcal{L}_{cyc} = \frac{1}{N} \sum_i \lambda\, |\dot{p}_i|, \tag{4}$$

where N is the batch size and i ranges over the latent dimension. We choose a regularization penalty of $\lambda = 0.01$ for all experiments.

4.3. Training loss

Our model is trained using the loss proposed in Tulyakov et al. (2018) amended with our cyclic-coordinate loss function (see equation 4). During training, an image discriminator network D_I predicts whether or not an input image is from the training set, and a video discriminator network D_V predicts whether or not a video is from the training set. The generator and discriminator networks are trained under the objective function:

$$\begin{aligned}\min_{G_I, R}\max_{D_I, D_V} \mathcal{L}(D_I, D_V, G_I, R) &= E_{\tilde{v}}[\log(1 - D_I(S_1(\tilde{v})))] + E_v[\log D_I(S_1(v))] \\ &+ E_{\tilde{v}}[\log(1 - D_V(S_T(\tilde{v})))] + E_v[\log D_V(S_T(v))] + \mathcal{L}_{cyc}\end{aligned}$$

where v denotes a video from the training dataset, $\tilde{v}$ denotes a synthetic video, and S_i denotes the random sampling of i consecutive video frames.

Table 1: FVD comparison. The consistent advantage of our model over MoCoGAN suggests that the HNN recurrent module is able to generate latent trajectories on the motion manifold more reliably than the GRU module. The advantage of our model over the HGN model suggests that the implicitly learned configuration space structure is a better representation of the actual configuration space than the Gaussian prior used in HGN. The best score is shown in bold, the second best is shown in blue.

	Mass spring	Pendulum	Double Pendulum	Two-body	Three-body
HGN	385.08	688.12	331.94	830.91	**451.40**
MoCoGAN	**31.37**	398.12	979.80	1028.13	2527.78
HGAN (Ours)	45.68	**91.64**	**73.21**	**105.85**	1981.10

5. Empirical analysis

In this section we compare videos generated using the following models:

1. HGN (Toth et al. (2019)): A VAE-based video generation approach in which the latent vector is interpreted as an element of the phase-space and propagated forward in time with an HNN (Greydanus et al. (2019)) cell. We use the pytorch implementation of HGN introduced in Rodas et al. (2021).

2. MoCoGAN (Tulyakov et al. (2018)): A GAN-based video generation approach in which content and motion vectors are initialized with Gaussian random noise. A motion sequence is generated using a discrete time RNN cell, and a latent sequence is constructed by concatenating the content vector to each motion vector. Each latent vector is decoded by the generator network to produce a sequence of images. We use the pytorch implementation of MoCoGAN introduced in Kitagawa and Kakiuch (2017).

3. HNN-GAN (Ours): This model differs from MoCoGAN in that the discrete time RNN cell is replaced with a continuous time HNN cell. This model can be seen as a variant of HGAN (our model), in which the configuration space map is ablated.

4. HGAN (Ours): This model differs from HNN-GAN in that a configuration space map transforms the Gaussian random noise initialized motion vector to an intermediate space. The resulting vector is interpreted as the initial condition for the HNN cell and propagated forward in time to give a motion sequence. Moreover, a cyclic-coordinate loss function is used during training to encourage a compact representation of the configuration-space.

All models are trained on the Hamiltonian Dynamics Suite Toy Physics dataset (Botev et al. (2021)). HGN and MoCoGAN are trained using the hyperparameters provided in the original paper. HNN-GAN and HGAN are trained using the hyperparameters proposed in MoCoGAN. The additional cyclic-coordinate loss regularizer in HGAN is found using grid search $\lambda \in [0.1, 0.01, 0.001]$. We find $\lambda = 0.01$ works well for our experiments.

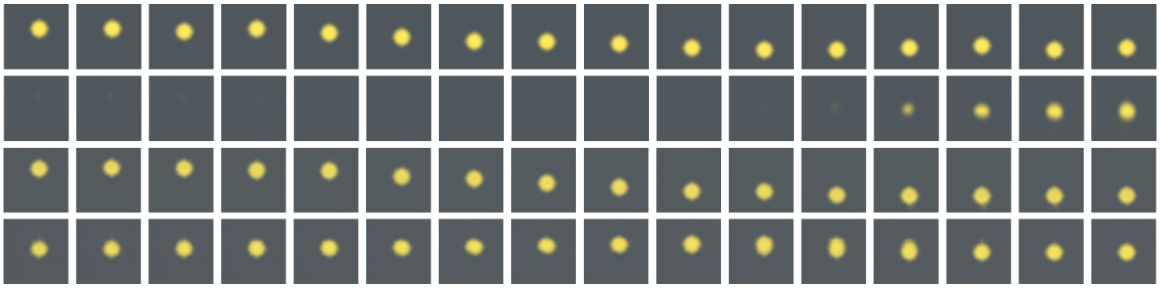

Figure 2: Unconditional mass spring video generation. Randomly sampled trajectories from the training set, HGN, MoCoGAN, and our model (top-bottom).

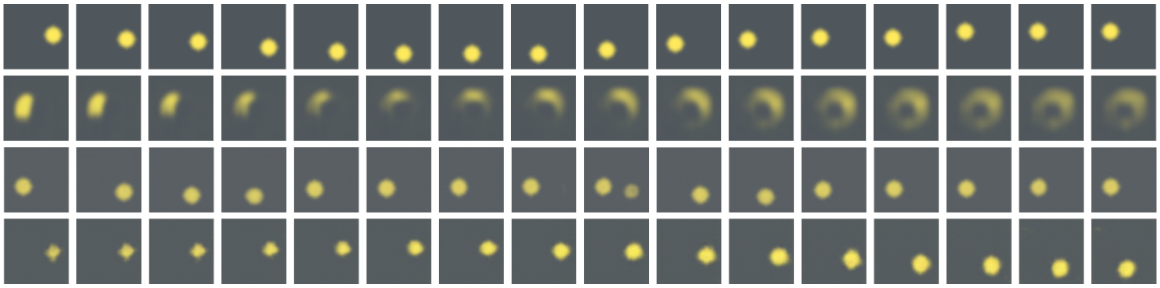

Figure 3: Unconditional pendulum video generation. Randomly sampled trajectories from the training set, HGN, MoCoGAN, and our model (top-bottom).

5.1. Datasets

We compare model performance on the Hamiltonian Dynamics Suite Toy Physics dataset (Botev et al. (2021)). The dataset consists of a collection of video renderings of simulated physical systems. It includes rendered simulations of the mass spring, double pendulum, pendulum, two-body and three-body systems. Simulations are performed with and without friction; and videos are rendered with constant or varied physical quantities (e.g., the mass of the pendulum bob, the length of the pendulum arm, and location of the pivot in the pendulum dataset) across trajectories, and constant grayscale, constant color, or varied color across trajectories. Each simulation is sampled at $dt = 0.05$, and 50k trajectories of 512 samples are generated for every system. Further details about the dataset can be found in (Botev et al. (2021)). Following HGN we evaluate all models on the version of this dataset generated without friction, with constant physical quantities across trajectories, and with constant color.

5.2. Perceptual quality

We compare the performance of our model against the baseline models both quantitatively and qualitatively. We quantitatively compare the perceptual quality of generated videos using the Fréchet Video Distance (FVD) (Unterthiner et al. (2019)). To avoid discrepancies in the evaluation due to subsampling and data processing procedures, we use FVD evaluation pipeline proposed in Sko-

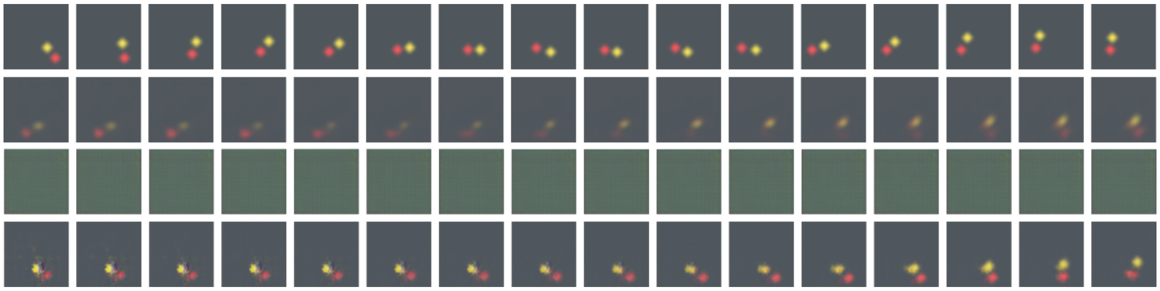

Figure 4: Unconditional double pendulum video generation. Randomly sampled trajectories from the training set, HGN, MoCoGAN, and our model (top-bottom).

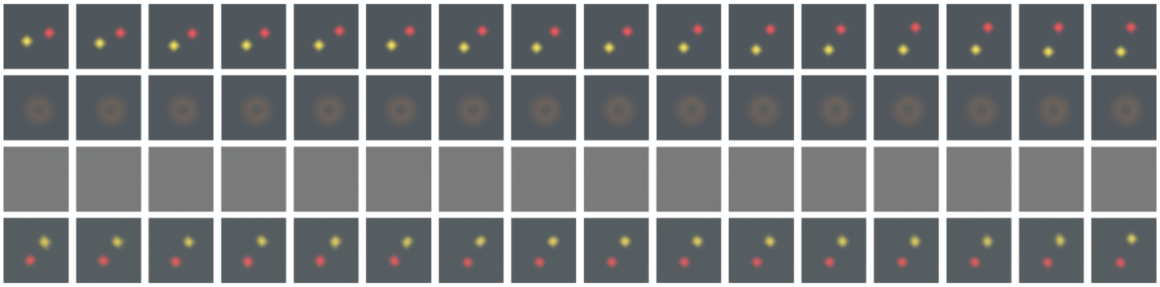

Figure 5: Unconditional two body video generation. Randomly sampled trajectories from the training set, HGN, MoCoGAN, and our model (top-bottom).

rokhodov et al. (2021), and compute the FVD score for each model using 2048 videos of 16 frames. We compare the perceptual quality of videos qualitatively by randomly generating 5 video sequences for each model and selecting the best for comparison. Our random seed is set to 0 for all models.

We compare the performance of our model against the HGN, and MoCoGAN baseline models. We randomly generate synthetic video sequences using each model and compare them in Figures 2-6; we report the FVD scores for each model in Table 1. Our model out performs the baseline models in three out of five cases (i.e., Pendulum, Double Pendulum, Two-body), and demonstrates comparable performance to the leading model (MoCoGAN) in one of the remaining cases (i.e., Mass Spring). There is a significant difference in the FVD score of the leading model (HGN), and our model on the Three-body case. As evidenced by Figure 6, our model produces qualitatively better videos than HGN. We attribute the discrepancy in FVD score to the difference in the background color of videos generated with our model and HGN. While videos generated with our model have a foreground more similar to that of the real data, those generated with HGN have a more similar background.

5.3. Configuration space structure

We investigate the impact of the configuration space map, and cyclic-coordinate loss on the learned latent structure. We sample 1024 Gaussian random motion vectors and propagate them through

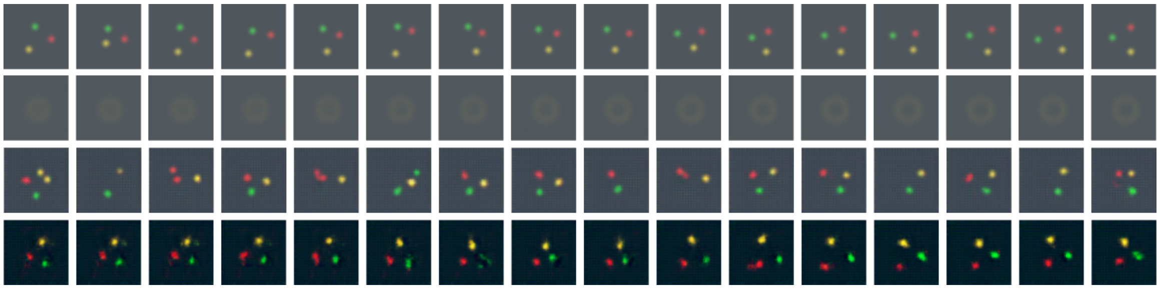

Figure 6: Unconditional three-body video generation. Randomly sampled trajectories from the training set, HGN, MoCoGAN, and our model (top-bottom).

the synthesis pipelines of MoCoGAN, HNN-GAN, and our model trained on the Pendulum in the Toy Physics dataset. We visualize the motion manifold using t-SNE (Van der Maaten and Hinton (2008)) projections in Figure 7. Visualizations of the MoCoGAN and HNN-GAN motion manifolds show no discernible structure, while the motion manifold for our model suggests the system is one dimensional.

6. Conclusion

In this work, we introduce a video generation model for physical systems that does not require an explicit prior on the structure of the configuration space. We learn a representation of the underlying configuration space, and leverage the Hamiltonian formalism to learn a continuous and conservative motion model. While the latent dimension of our model exceeds the dimension of the systems considered, we are able to learn compact representations of the configuration space because of our physics-inspired cyclic-coordinate loss function. Our model outperforms baselines in many cases, and affords a mechanism for investigating the learned configuration space structure.

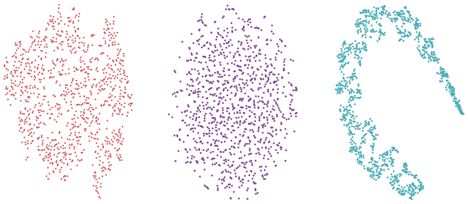

Figure 7: t-SNE projection of the motion manifold. We propagate the Gaussian random motion vector through the synthesis pipelines of MoCoGAN, HNN-GAN, and our model (left-right), and visualize the one step output of the motion module.

Acknowledgments

The authors gratefully acknowledge support from Princeton SEAS Innovation Grant.

References

Christine Allen-Blanchette, Sushant Veer, Anirudha Majumdar, and Naomi Ehrich Leonard. Lagnetvip: A lagrangian neural network for video prediction. *arXiv preprint arXiv:2010.12932*, 2020.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. Which priors matter? benchmarking models for learning latent dynamics. 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. *arXiv preprint arXiv:1909.13334*, 2019.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Herbert Goldstein, Charles Poole, and John Safko. Classical mechanics, 2002.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Cade Gordon and Natalie Parde. Latent neural differential equations for video generation. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pages 73–86. PMLR, 2021.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

Irina Higgins, Peter Wirnsberger, Andrew Jaegle, and Aleksandar Botev. Symetric: Measuring the quality of learnt hamiltonian dynamics inferred from vision. *Advances in Neural Information Processing Systems*, 34, 2021.

Darryl D Holm, Tanya Schmah, and Cristina Stoica. *Geometric mechanics and symmetry: from finite to infinite dimensions*, volume 12. Oxford University Press, 2009.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Shingo Kitagawa and Kota Kakiuch. A pytorch implmention of MoCoGAN. https://github.com/DLHacks/mocogan, 2017.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

Carles Balsells Rodas, Oleguer Canal, and Federico Taschin. Re-hamiltonian generative networks. In *ML Reproducibility Challenge 2020*, 2021.

Manuel A Roehrl, Thomas A Runkler, Veronika Brandtstetter, Michel Tokic, and Stefan Obermayer. Modeling system dynamics with physics-informed neural networks based on lagrangian mechanics. *IFAC-PapersOnLine*, 53(2):9195–9200, 2020.

Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Deisenroth. Variational integrator networks for physically structured embeddings. In *International Conference on Artificial Intelligence and Statistics*, pages 3078–3087. PMLR, 2020.

Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *arXiv preprint arXiv:2112.14683*, 2021.

Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021.

Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016.

Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G3an: Disentangling appearance and motion for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5264–5273, 2020.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571*, 2022.

Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. *Advances in Neural Information Processing Systems*, 33, 2020.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.