# Discovering Phase Space Structure in Learned Hamiltonian Systems

**Yulong Yang**[*,†] **Jiayin Liu**[*] **Vineet Bansal** **Christine Allen-Blanchette**[†]
Princeton University
{yulong.yang, jl4266, vineetb, ca15}@princeton.edu

## Abstract

Mechanics-inspired neural networks provide physically structured models of dynamical systems, but typically assume access to the coordinates in which the underlying dynamics are expressed. While recent generative approaches relax this assumption, they remain limited in their ability to jointly model different dynamical regimes and discover interpretable latent structure. We introduce Structured Phase Space GAN (SPS-GAN), a conditional generative framework that learns mechanical dynamics and discovers latent phase-space structure from Cartesian trajectories or video observations. We achieve this by combining adversarial generation with port-Hamiltonian latent dynamics and extending cyclic-coordinate regularization to account for dissipation and forcing. This formulation supports conservative, dissipative, and forced systems while encouraging compact latent representations from which active degrees of freedom can be identified. Experiments on simulated and real-world systems demonstrate improved trajectory accuracy and video generation quality over competitive baselines, while identifying active latent dimensions consistent with the expected effective degrees of freedom and recovering physically interpretable coordinates across observation modalities and dynamical regimes.

## 1 Introduction

Learning models of dynamical systems from observations is fundamental to applications in robotics and control [1], autonomous driving [2], and aerospace engineering [3]. However, learning physically interpretable models remains challenging because the generalized coordinates that describe the underlying mechanical structure are often not directly observable. Although observations may consist of Cartesian trajectories or video frames, the underlying motion may admit a lower-dimensional mechanical representation that reflects the system's constraints and symmetries. Classical system identification methods learn dynamical models from observations [4, 5, 6, 7, 8, 9], but often rely on prescribed model structures or assumptions about the underlying state representation. This motivates learning models that not only reproduce observed dynamics but also discover compact, physically interpretable representations of the underlying motion.

Recent neural network approaches have incorporated principles from classical mechanics to improve the physical consistency and interpretability of learned dynamics. Hamiltonian and Lagrangian neural networks learn energy functions whose induced equations of motion provide structured predictions of system evolution [10, 11, 12]. Extensions to dissipative and forced systems broaden the class of dynamics that can be modeled [13, 14, 15]. Mechanics-inspired generative models further learn distributions over trajectories or videos using structured latent dynamics, including Hamiltonian Generative Networks [16] and Hamiltonian GAN [17]. In particular, Hamiltonian GAN introduces cyclic-coordinate regularization to encourage compact latent phase-space representations, but is formulated for conservative dynamics without conditional generation across systems. Complementary

[*] Equal contribution. [†] Corresponding author. This work was done while Jiayin Liu was a visiting student at Princeton.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

approaches identify conserved quantities and symmetries directly from data [18, 19, 20], but generally focus on identifying dynamical invariants rather than learning generative models.

Despite these advances, learning generative models that capture physically plausible motion while discovering interpretable latent mechanical structure remains challenging. We introduce Structured Phase Space GAN (SPS-GAN), a conditional generative framework that addresses this challenge using Cartesian trajectories or video observations. The central idea is to constrain the generative process through structured latent dynamics, enabling mechanical structure to be identified from the learned phase space. SPS-GAN maps sampled motion codes and system information to initial latent phase-space states, which evolve according to port-Hamiltonian dynamics. The resulting trajectories are decoded into the observation space. The port-Hamiltonian formulation supports conservative, dissipative, and forced dynamics, while adversarial training encourages the generated motions to match the observed distribution. To encourage compact latent structure, we adapt cyclic-coordinate regularization [17] to account for dissipation and forcing, encouraging redundant coordinate–momentum pairs to become inactive. Analyzing and decoding the learned latent representation then allows us to identify active dimensions and interpret their relationship to physical mechanical coordinates.

We evaluate SPS-GAN on simulated systems spanning conservative, dissipative, and forced dynamics, as well as real-world trajectory and video datasets. SPS-GAN achieves competitive or improved trajectory accuracy relative to supervised dynamics baselines and consistently improves video generation quality over generative baselines. Analysis of the learned phase space further reveals active latent dimensions consistent with the expected effective degrees of freedom and physically interpretable coordinates across the evaluated systems. Our main contributions are:

- We introduce SPS-GAN, a conditional generative framework for learning physically structured dynamics from Cartesian trajectories or video observations, supporting generation across systems and parameter variations.
- We extend Hamiltonian generative modeling to dissipative and forced systems through port-Hamiltonian latent dynamics, adapting cyclic-coordinate regularization to encourage compact latent representations.
- We demonstrate that SPS-GAN identifies compact latent representations whose active dimensions match the expected effective degrees of freedom of the evaluated systems, with decoded coordinates corresponding to interpretable mechanical motion.

## 2 Related Work

**Hamiltonian Mechanics in Neural Networks.** Hamiltonian inductive biases incorporate principles of classical mechanics into neural dynamical models, providing physically structured representations of system evolution. Hamiltonian Neural Networks (HNNs) [10] learn a system Hamiltonian and predict dynamics using Hamilton's equations. Subsequent approaches incorporate symplectic integration, controlled dynamics, and geometric structure to improve physical consistency and long-horizon prediction [11, 21, 22, 23, 24]. Port-Hamiltonian neural networks extend Hamiltonian modeling to dissipative and forced systems by explicitly separating energy-conserving dynamics from dissipation and external forcing [13, 15]. Mechanics-inspired inductive biases have also been applied to high-dimensional observations, including video prediction of physical dynamics [25, 26]. In contrast to these approaches, SPS-GAN uses port-Hamiltonian latent dynamics within a conditional generative framework to model distributions over physically plausible motions.

**Generating Physically Plausible Data.** Mechanics-inspired generative models incorporate physical inductive biases to synthesize trajectories and videos with structured dynamics. Hamiltonian Generative Networks (HGNs) [16] generate videos using learned latent Hamiltonian dynamics and symplectic integration. Subsequent approaches explore neural differential equations for video generation [27] and Hamiltonian latent operators for disentangling motion and appearance [28]. Hamiltonian GAN (HGAN) [17] combines adversarial generation with Hamiltonian latent dynamics to synthesize physically plausible motion. Beyond generating physically plausible motion, SPS-GAN also discovers compact, interpretable latent mechanical structure from observations.

**Discovering Structure in Dynamical Systems.** Complementary approaches seek to discover dynamical structure directly from observations through conserved quantities and symmetries. COMET [18] learns candidate constants of motion by enforcing conservation along observed trajectories, while FINDE [19] learns first integrals and incorporates them into neural differential equations to preserve

the identified invariants during prediction. Other approaches investigate symmetry structure through prescribed group actions: SymDLNN [20] identifies symmetry subgroups, while Latent MOS [29] incorporates predefined symmetry groups into a mixture-of-experts architecture. These methods focus on identifying invariants or symmetries rather than discovering compact latent representations within a generative model. Hamiltonian GAN [17] addresses the latter objective by introducing cyclic-coordinate regularization to encourage compact latent Hamiltonian representations. SPS-GAN extends this regularization principle to port-Hamiltonian dynamics by accounting for dissipation and forcing, enabling latent structure discovery across conservative and non-conservative systems.

# 3 Background

We review Hamiltonian mechanics and port-Hamiltonian neural networks, which provide the mathematical foundations for SPS-GAN. In particular, cyclic coordinates motivate the latent structure-discovery regularizer, while port-Hamiltonian dynamics enable modeling of conservative, dissipative, and forced systems.

## 3.1 Hamiltonian Mechanics

Hamiltonian mechanics describes conservative mechanical systems in terms of generalized coordinates $\mathbf{q}$ and their conjugate momenta $\mathbf{p}$ [30]. Together, these variables define the phase-space state $\mathbf{z} = (\mathbf{q}, \mathbf{p})$. For the systems considered here, the Hamiltonian $\mathcal{H}(\mathbf{q}, \mathbf{p})$ represents the total mechanical energy, and the dynamics follow Hamilton's equations,

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix} \nabla \mathcal{H}(\mathbf{q}, \mathbf{p}). \tag{1}$$

A generalized coordinate $q_k$ is cyclic, or ignorable, if the Hamiltonian is independent of that coordinate, i.e., $\partial \mathcal{H}/\partial q_k = 0$. Hamilton's equations then imply conservation of its conjugate momentum,

$$\dot{p}_k = -\partial \mathcal{H}/\partial q_k = 0. \tag{2}$$

This property motivates the cyclic-coordinate regularization used in Section 4.2 to encourage compact latent phase-space representations.

## 3.2 Port Hamiltonian Neural Networks

Standard autonomous Hamiltonian dynamics conserve energy and do not directly account for dissipation or external forcing. The port-Hamiltonian formulation [31] extends Hamiltonian dynamics to incorporate these effects,

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \left( \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{D} \end{bmatrix} \right) \nabla \mathcal{H}(\mathbf{q}, \mathbf{p}) + \begin{bmatrix} 0 \\ \mathbf{G}(\mathbf{q}) \end{bmatrix} \mathbf{u}, \tag{3}$$

Here, $\mathbf{D} \succeq 0$ captures energy dissipation, $\mathbf{u}$ denotes an external input, and $\mathbf{G}(\mathbf{q})$ maps that input into the momentum dynamics. This formulation separates energy-conserving dynamics from dissipative effects and external forcing.

For systems with unknown dynamics, Hamiltonian Neural Networks (HNNs) [10] parameterize the Hamiltonian using a neural network and derive the dynamics through Hamilton's equations. Port-Hamiltonian Neural Networks (pHNNs) [15] extend this approach by learning the Hamiltonian alongside dissipation and forcing terms. In Section 4.2, we use a pHNN to evolve learned phase-space states, without requiring access to prescribed canonical coordinates.

# 4 Structured Phase Space GAN

Structured Phase Space GAN (SPS-GAN) is a conditional generative framework for learning physically structured dynamics from either Cartesian trajectories or video observations, as illustrated in Figure 1. SPS-GAN maps sampled motion codes and system information to initial latent phase-space states, which evolve according to port-Hamiltonian dynamics before being decoded into observations. Adversarial training encourages the generated sequences to match the observed data distribution, while cyclic-coordinate regularization encourages compact latent phase-space representations.

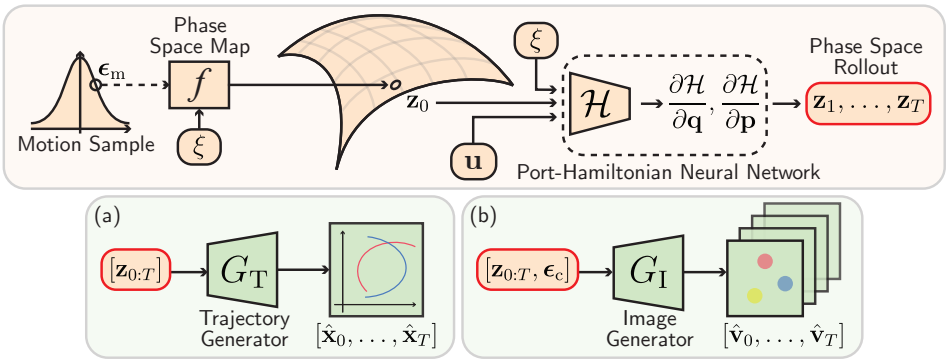



Figure 1: **SPS-GAN generates trajectories and videos across multiple systems through a shared phase-space backbone.** Random motion samples $\boldsymbol{\epsilon}_{\mathrm{m}}$ and system information $\xi$ are passed through the phase-space map $f$ to produce an initial phase-space state $\mathbf{z}_0$. This state is evolved forward in time by a Port-Hamiltonian Neural Network [15] conditioned on $\xi$, yielding the phase-space trajectory $[\mathbf{z}_{0:T}]$, which is decoded to **(a)** Cartesian trajectory $[\hat{\mathbf{x}}_{0:T}]$ and **(b)** video frames $[\hat{\mathbf{v}}_{0:T}]$.

### 4.1 Phase-Space Map

Port-Hamiltonian dynamics provide a physically structured model of system evolution, but require an initial phase-space state. Since SPS-GAN generates trajectories without prescribed initial conditions or canonical coordinates, we learn a conditional map from sampled motion codes to latent initial states. Let $\xi$ denote system information, such as system identity or physical parameters, and let $\boldsymbol{\epsilon}_{\mathrm{m}} \sim \mathcal{N}(0, \mathbf{I}_{d_{\mathrm{m}}})$ denote a random motion code. An MLP $f$ maps these inputs to an initial latent phase-space state,

$$\mathbf{z}_0 = f(\boldsymbol{\epsilon}_{\mathrm{m}}, \xi) \in \mathbb{R}^{2d_{\mathrm{lat}}}, \tag{4}$$

where $\mathbf{z}_0 = (\mathbf{q}_0, \mathbf{p}_0)$ consists of learned generalized coordinates $\mathbf{q}_0 \in \mathbb{R}^{d_{\mathrm{lat}}}$ and conjugate momenta $\mathbf{p}_0 \in \mathbb{R}^{d_{\mathrm{lat}}}$. The map induces a conditional distribution over latent initial states, enabling diverse trajectory generation for a given system. We choose $d_{\mathrm{lat}}$ as an upper bound on the expected effective dimensionality of the observed dynamics (see Appendix C.1 for sensitivity analysis). To encourage a compact representation within this potentially over-parameterized space, we introduce cyclic-coordinate regularization in Section 4.2.

### 4.2 Phase-Space Dynamics

Starting from the initial state $\mathbf{z}_0$ produced by the phase-space map, SPS-GAN generates a latent trajectory $\mathbf{z}_{0:T} = [\mathbf{z}_0, \ldots, \mathbf{z}_T]$ using a pHNN conditioned on system information $\xi$,

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \int_t^{t+1} \left( \left( \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{D}_\phi(\xi) \end{bmatrix} \right) \nabla_{\mathbf{z}} \mathcal{H}_\theta(\mathbf{z}_\tau, \xi) + \begin{bmatrix} 0 \\ \mathbf{F}_\psi(\mathbf{u}_\tau, \xi) \end{bmatrix} \right) d\tau. \tag{5}$$

Here, $\mathcal{H}_\theta(\mathbf{z}, \xi)$ is the learned Hamiltonian, while $\mathbf{D}_\phi(\xi)$ and $\mathbf{F}_\psi(\mathbf{u}, \xi)$ model dissipation and external forcing, respectively. The dissipation matrix is constrained to be positive semi-definite, as described in Appendix B.1. For conservative systems, setting the non-conservative terms to zero recovers standard Hamiltonian neural dynamics.

When the latent phase space is over-parameterized, learned dynamics may occupy more coordinates than are needed to represent the observed motion. To encourage compact latent representations, we adapt the cyclic-coordinate regularization introduced in [17]. In conservative Hamiltonian systems, a cyclic coordinate has a conserved conjugate momentum. In port-Hamiltonian systems, however, dissipation and forcing can change the conjugate momentum even when the corresponding coordinate is cyclic. From Equation (5), the momentum dynamics are

$$\dot{\mathbf{p}} = -\nabla_{\mathbf{q}} \mathcal{H}_\theta(\mathbf{z}, \xi) - \mathbf{D}_\phi(\xi) \nabla_{\mathbf{p}} \mathcal{H}_\theta(\mathbf{z}, \xi) + \mathbf{F}_\psi(\mathbf{u}, \xi). \tag{6}$$

To isolate the Hamiltonian contribution from dissipation and forcing, we define the cyclic-coordinate loss

$$\mathcal{L}_{\mathrm{cyclic}} = \lambda_{\mathrm{cyclic}} \sum_{i=1}^{d_{\mathrm{lat}}} \left| \dot{p}_i + [\mathbf{D}_\phi(\xi) \nabla_{\mathbf{p}} \mathcal{H}_\theta(\mathbf{z}, \xi) - \mathbf{F}_\psi(\mathbf{u}, \xi)]_i \right|. \tag{7}$$

By Equation (6), this loss is equivalent to penalizing the coordinate derivatives of the Hamiltonian, $\lambda_{\text{cyclic}} \sum_i |\partial \mathcal{H}_\theta / \partial q_i|$. It therefore encourages the learned Hamiltonian to become independent of selected coordinates without penalizing momentum changes arising from non-conservative effects. Following [15], we additionally regularize the dissipation and forcing terms,

$$\mathcal{L}_{\text{non-cons}} = \lambda_D \|\mathbf{D}_\phi(\xi)\| + \lambda_F \|\mathbf{F}_\psi(\mathbf{u}, \xi)\|. \tag{8}$$

This penalty discourages unnecessary non-conservative contributions, while cyclic-coordinate regularization encourages compact latent mechanical representations. Cyclic-coordinate regularization is intended to encourage compact latent representations. We evaluate the resulting active dimensionality and mechanical interpretability in Sections 5.3 and 5.4.

### 4.3 Generating Cartesian Trajectories

To generate Cartesian trajectories, SPS-GAN decodes each latent phase-space state using an MLP generator $G_{\text{T}}$,

$$\tilde{\mathbf{x}}_t = G_{\text{T}}(\mathbf{z}_t) \in \mathbb{R}^{d_{\text{out}}}, \tag{9}$$

where $d_{\text{out}}$ is the maximum flattened Cartesian observation dimension across the systems in the dataset. To accommodate systems with different numbers of particles, we apply a system-specific binary mask $\mathbf{M}(\xi)$,

$$\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t \odot \mathbf{M}(\xi), \tag{10}$$

which removes unused output coordinates. The generated trajectory is $\hat{\mathbf{x}}_{0:T} = [\hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_T]$, and real trajectories are padded to the same dimension.

We train a recurrent discriminator $D_{\text{T}}(\cdot, \xi))$ to distinguish generated trajectory subsequences from real observations, conditioned on system information $\xi$. The conditional adversarial objective is

$$V_{\text{traj}}(G_{\text{T}}, D_{\text{T}}) = \mathbb{E}_{\mathbf{x} \sim p_X(\cdot|\xi)} \left[\log D_{\text{T}}(\mathbf{x}_{s:s+\ell}, \xi)\right] + \mathbb{E}_{\hat{\mathbf{x}} \sim p_{G_{\text{T}}(\cdot|\xi)}} \left[\log\left(1 - D_{\text{T}}(\hat{\mathbf{x}}_{s:s+\ell}, \xi)\right)\right]. \tag{11}$$

The generator, phase-space map, and pHNN are trained jointly using the adversarial objective and the cyclic-coordinate and non-conservative regularization terms introduced in Section 4.2. We evaluate the dynamical consistency of the generated Cartesian trajectories against supervised prediction baselines in Section 5.1.

### 4.4 Generating Videos

SPS-GAN uses the same phase-space map, latent dynamics, and cyclic-coordinate regularization across observation modalities, adapting only the observation generator and discriminators. For video observations, a convolutional generator $G_{\text{I}}$ decodes each latent state $\mathbf{z}_t$ into an RGB frame. To model appearance variation separately from motion, we sample a content code $\boldsymbol{\epsilon}_{\text{c}} \sim \mathcal{N}(0, \mathbf{I}_{d_{\text{c}}})$ and hold it fixed across the generated sequence. The generator produces

$$\hat{\mathbf{v}}_t = G_{\text{I}}([\mathbf{z}_t, \boldsymbol{\epsilon}_{\text{c}}]) \in \mathbb{R}^{d_{\text{h}} \times d_{\text{w}} \times 3}, \tag{12}$$

yielding the video sequence $\hat{\mathbf{v}}_{0:T} = [\hat{\mathbf{v}}_0, \ldots, \hat{\mathbf{v}}_T]$.

To encourage both visual realism and temporal coherence, we use two discriminators conditioned on system information $\xi$. The image discriminator $D_{\text{I}}(\cdot, \xi)$ evaluates individual frames, while the video discriminator $D_{\text{V}}(\cdot, \xi)$ evaluates sampled subsequences $\mathbf{v}_{s:s+\ell}$. The corresponding adversarial objective is

$$\begin{aligned} V_{\text{video}}(G_{\text{I}}, D_{\text{I}}, D_{\text{V}}) = {} & \mathbb{E}_{\mathbf{v} \sim p_V(\cdot|\xi)} \left[\log D_{\text{V}}(\mathbf{v}_{s:s+\ell}, \xi) + \frac{1}{T} \sum\nolimits_{t=0}^{T} \log D_{\text{I}}(\mathbf{v}_t, \xi)\right] \\ & + \mathbb{E}_{\hat{\mathbf{v}} \sim p_{G_{\text{I}}}(\cdot|\xi)} \left[\log\left(1 - D_{\text{V}}(\hat{\mathbf{v}}_{s:s+\ell}, \xi)\right) + \frac{1}{T} \sum\nolimits_{t=0}^{T} \log\left(1 - D_{\text{I}}(\hat{\mathbf{v}}_t, \xi)\right)\right]. \end{aligned} \tag{13}$$

The video generator, phase-space map, and pHNN are trained jointly using the corresponding generator loss and the regularization terms $\mathcal{L}_{\text{cyclic}}$ and $\mathcal{L}_{\text{non-cons}}$. We evaluate the visual fidelity and temporal consistency of the generated sequences in Section 5.2.

# 5 Results

We evaluate SPS-GAN along two dimensions, focusing on the dynamical fidelity of generated motion and the dimensionality and interpretability of the learned latent phase space. Experiments span ordered, chaotic, dissipative, and forced systems using simulated and real-world Cartesian trajectories and video observations. We assess dynamical fidelity through initial-condition-aligned comparisons with supervised trajectory prediction models, and evaluate video quality and temporal consistency against generative baselines. We further examine whether the learned representations exhibit the expected effective dimensionality and whether their active directions correspond to interpretable mechanical coordinates.

## 5.1 Generating Cartesian Trajectories

We first evaluate the dynamical consistency of trajectories generated by SPS-GAN. Unlike predictive baselines that evolve prescribed initial conditions, SPS-GAN samples trajectories from a learned distribution, requiring an initial-condition alignment protocol for comparison. For each generated trajectory $\hat{\mathbf{x}}_{0:T}$, we initialize the true simulator and predictive baselines at $\hat{\mathbf{x}}_0$, then compute the MSE of each rollout against the simulator trajectory. This evaluates whether SPS-GAN generates trajectories consistent with the underlying dynamics rather than reconstructing particular observed trajectories.

We evaluate seven systems, namely mass-spring, single pendulum, double pendulum, two-body, three-body, damped pendulum, and forced Duffing, which spans conservative, dissipative, forced, and chaotic dynamics. Details on training data and example rollouts are provided in Appendices A and B.1.

**Conservative Systems.** We evaluate whether SPS-GAN can accurately capture conservative dynamics under both single-system and joint conditional training. We consider two configurations: SPS-GAN, trained independently on each system, and cSPS-GAN, a single conditional model trained jointly across all five systems using system labels. For both configurations, we set $\mathbf{D}_\phi = \mathbf{F}_\psi = 0$. We compare against supervised dynamics baselines HNN [10], SymODEN [21], NeuralODE [32], LatentODE [33], and TRS-ODEN [34], and generative baselines HGN [16], TimeGAN [35], Koopman VAE [36], and DiffusionTS [37]. As shown in Table 1, SPS-GAN achieves the lowest MSE on four of the five systems. cSPS-GAN achieves comparable accuracy on most systems while modeling all five systems jointly. These results indicate that SPS-GAN can generate dynamically consistent trajectories without prescribed initial conditions, and that conditional training enables multiple systems to share a single model.

**Long-Horizon Distributional Fidelity.** In chaotic systems, pointwise trajectory errors can grow rapidly and do not necessarily reflect agreement in long-term motion statistics [38, 39]. We therefore evaluate the long-horizon distributional fidelity of SPS-GAN on the chaotic double pendulum. We train all models with a rollout horizon of $T = 200$ and compare 4,096 generated rollouts with 4,096

Table 1: **Trajectory generation accuracy on conservative systems.** We report MSE between each generated or predicted trajectory and the simulator rollout initialized from the same state. SPS-GAN is trained per system, while cSPS-GAN is trained jointly across all five systems with system labels. Best results are bolded and second-best results are underlined.

| Network | Mass Spring | Single Pend. | Double Pend. | Two Body | Three Body |
|---|---|---|---|---|---|
| SPS-GAN | $6.16 \times 10^{-4}$ | $\mathbf{4.29 \times 10^{-4}}$ | $\mathbf{3.25 \times 10^{-2}}$ | $\mathbf{3.07 \times 10^{-3}}$ | $\mathbf{1.26 \times 10^{-3}}$ |
| HNN | $3.53 \times 10^{-4}$ | $6.27 \times 10^{-4}$ | $1.25 \times 10^{-1}$ | $\underline{3.61 \times 10^{-3}}$ | $2.81 \times 10^{-3}$ |
| SymODEN | $6.40 \times 10^{-4}$ | $6.56 \times 10^{-4}$ | $9.89 \times 10^{-2}$ | $6.20 \times 10^{-3}$ | $8.08 \times 10^{-3}$ |
| TRS-ODEN | $6.72 \times 10^{-4}$ | $9.57 \times 10^{-4}$ | $6.12 \times 10^{-2}$ | $6.36 \times 10^{-3}$ | $6.22 \times 10^{-3}$ |
| NeuralODE | $\underline{1.62 \times 10^{-4}}$ | $1.59 \times 10^{-3}$ | $5.99 \times 10^{-2}$ | $5.72 \times 10^{-3}$ | $3.67 \times 10^{-3}$ |
| LatentODE | $\mathbf{1.13 \times 10^{-4}}$ | $9.27 \times 10^{-4}$ | $4.66 \times 10^{-2}$ | $1.98 \times 10^{-2}$ | $\underline{1.37 \times 10^{-3}}$ |
| HGN | $1.81 \times 10^{-3}$ | $1.92 \times 10^{-2}$ | $2.27 \times 10^{-1}$ | $9.31 \times 10^{-1}$ | $7.99 \times 10^{-2}$ |
| TimeGAN | $1.17 \times 10^{-2}$ | $2.43 \times 10^{-2}$ | $5.82 \times 10^{-1}$ | $1.21 \times 10^{0}$ | $1.97 \times 10^{0}$ |
| Koopman VAE | $6.84 \times 10^{-4}$ | $5.27 \times 10^{-3}$ | $5.46 \times 10^{-2}$ | $9.36 \times 10^{-3}$ | $1.13 \times 10^{-2}$ |
| DiffusionTS | $7.54 \times 10^{-4}$ | $5.52 \times 10^{-3}$ | $5.90 \times 10^{-2}$ | $3.83 \times 10^{-3}$ | $1.22 \times 10^{-2}$ |
| cSPS-GAN | $9.88 \times 10^{-4}$ | $\underline{5.11 \times 10^{-4}}$ | $\underline{3.33 \times 10^{-2}}$ | $4.46 \times 10^{-3}$ | $2.61 \times 10^{-3}$ |

Table 2: **Long-horizon distributional fidelity on the chaotic double pendulum.** We compare generative and deterministic models using phase-space distribution distances, energy-distribution agreement, relative energy drift, and Poincaré-section recurrence statistics. Lower values indicate better performance.

| Network | Type | Phase Space | | | Energy | | Poincaré |
|---|---|---|---|---|---|---|---|
| | | SWD | EMD | KL | SWD | Drift | SWD |
| SPS-GAN | Gen. | **0.054** | **0.256** | **0.370** | **0.056** | **4.521%** | **0.661** |
| TimeGAN | Gen. | 0.126 | 0.397 | 1.146 | 0.287 | 25.113% | 1.588 |
| Koopman VAE | Gen. | 0.281 | 0.584 | 0.713 | 0.501 | 15.751% | 1.853 |
| Diff-TS | Gen. | <u>0.090</u> | <u>0.305</u> | <u>0.554</u> | 0.472 | 29.203% | 1.643 |
| HGN | Gen. | 0.147 | 0.349 | 0.679 | 0.206 | 9.391% | 0.979 |
| NeuralODE | Det. | 0.329 | 0.667 | 0.994 | 0.507 | 24.721% | 1.327 |
| LatentODE | Det. | 0.241 | 0.544 | 0.561 | 0.357 | 23.608% | 1.134 |
| SymODEN | Det. | 0.212 | 0.707 | 2.563 | <u>0.144</u> | <u>7.159%</u> | 0.938 |
| HNN | Det. | 0.393 | 0.992 | 2.504 | 0.181 | 11.611% | <u>0.785</u> |

simulated reference rollouts. We compare the distributions of visited phase-space states using sliced Wasserstein distance (SWD), Earth mover's distance (EMD), and forward Kullback–Leibler (KL) divergence. We further assess energy-distribution agreement, energy conservation, and recurrence statistics using energy SWD, per-rollout energy drift, and Poincaré-section SWD, respectively. As shown in Table 2, SPS-GAN achieves the lowest values across all six metrics among the evaluated baselines. These results indicate that SPS-GAN captures complementary aspects of the system's long-term behavior, including phase-space occupancy, energy statistics, and recurrence structure.

**Non-Conservative Systems.** We next evaluate whether the port-Hamiltonian formulation supports accurate generation of dissipative and externally forced dynamics. We evaluate SPS-GAN on a damped pendulum and forced Duffing oscillator, with the learned dissipation and forcing components of the model enabled. As shown in Table 3, SPS-GAN achieves the lowest MSE on both systems when compared against pHNN, NeuralODE, LatentODE, and the dissipative variant of TRS-ODEN. We additionally evaluate an ideal pendulum as a conservative control, where SPS-GAN also achieves the lowest MSE with the full port-Hamiltonian parameterization. These results indicate that the framework accommodates non-conservative dynamics while retaining accuracy when dissipation and forcing are unnecessary. Additional experiments on an actuated pendulum with external control inputs are provided in Appendix B.3.

### 5.2 Generating Video Sequences

We evaluate whether SPS-GAN generates visually realistic videos with temporally consistent, physically plausible motion. Experiments consider rendered conservative systems, conditional generation under physical and appearance variation, and real-world single- and double-pendulum videos. Details on data generation and training are provided in Appendix A, with example rollouts in Appendix B.2.

**Fixed-Parameter Conservative Systems.** We first evaluate the visual fidelity and physical consistency of videos generated from conservative systems. SPS-GAN is trained separately on rendered observations of five conservative systems, without access to their underlying physical states. As shown in Table 4, SPS-GAN achieves the lowest Fréchet Video Distance (FVD) [40] on all five systems, outperforming generative baselines HGN [16] and HGAN [17]. Because FVD alone does

Table 3: **Trajectory generation accuracy on non-conservative systems.** We report MSE under the same initial-condition alignment protocol as Table 1. SPS-GAN uses the port-Hamiltonian formulation for dissipative and forced systems; the ideal pendulum is used as a conservative control.

| Network | Ideal Pend. | Damped Pend. | Forced Duffing |
|---|---|---|---|
| SPS-GAN | $\mathbf{4.77\times10^{-4}}$ | $\mathbf{4.57\times10^{-4}}$ | $\mathbf{5.22\times10^{-4}}$ |
| pHNN | <u>$8.42 \times 10^{-4}$</u> | <u>$4.76 \times 10^{-4}$</u> | $6.92 \times 10^{-4}$ |
| NeuralODE | $1.59 \times 10^{-3}$ | $6.85 \times 10^{-4}$ | $7.35 \times 10^{-4}$ |
| LatentODE | $9.27 \times 10^{-4}$ | $6.69 \times 10^{-4}$ | $1.18 \times 10^{-3}$ |
| TRS-ODEN | $9.57 \times 10^{-4}$ | $1.44 \times 10^{-3}$ | <u>$6.50 \times 10^{-4}$</u> |

Table 4: **Video generation quality and physical consistency.** We report FVD against baselines and energy variance ($\Delta E$) of centroid-extracted trajectories. SPS-GAN achieves the lowest FVD on all rendered conservative systems, while its $\Delta E$ closely matches the simulator reference floor.

| | Network | Mass Spring | Single Pend. | Double Pend. | Two Body | Three Body |
|---|---|---|---|---|---|---|
| FVD | SPS-GAN | **25.6±3.8** | **40.6±6.9** | **24.1±3.8** | **87.1±7.9** | **89.1±11.4** |
| | HGAN | 45.7±3.8 | 91.6±8.2 | 73.2±7.1 | 105.9±8.4 | 1981.1±98.6 |
| | HGN | 385.1±26.8 | 688.1±85.6 | 331.9±41.2 | 830.9±72.3 | 451.4±60.3 |
| $\Delta E$ | SPS-GAN | 2.33% | 2.89% | 6.83% | 5.96% | 3.47% |
| | Simulator | 1.64% | 2.62% | 6.32% | 6.11% | 3.57% |

not establish physical consistency, we extract particle centroids from generated videos and measure the energy variance along each rollout. We apply the same procedure to simulator videos, whose nonzero energy variance reflects errors introduced by centroid extraction rather than physical energy drift. SPS-GAN produces energy variation comparable to this simulator reference across all five systems, providing evidence that its generated videos approximately preserve the expected conservative behavior.

**Variable-Parameter Conservative Systems.** We next evaluate whether a single conditional SPS-GAN model can generate videos across multiple systems while accommodating variation in physical and visual parameters. We train cSPS-GAN jointly across systems using conditional labels $\xi$ specifying system identity and parameter settings, including particle size and color. Figure 6 in Appendix B.2 illustrates generated rollouts under different physical and visual conditions. cSPS-GAN achieves FVDs of 135.3–182.2, outperforming the single-system HGN results reported in Table 4. These results demonstrate that a single conditional model can generate videos across multiple systems and parameter configurations.

**Real-World Systems.** We next evaluate whether SPS-GAN generates dynamically consistent videos from real-world observations, where non-conservative effects and complex visual appearances depart from idealized simulations. We train SPS-GAN on real-world videos of a freely swinging single pendulum [41] and double pendulum [42]. Figure 2 shows that SPS-GAN generates temporally coherent pendulum motion, whereas HGN produces nearly static sequences and HGAN exhibits visibly inconsistent motion. For the single pendulum, the energy variance of the centroid-extracted SPS-GAN videos is 2.9%, closely matching the 2.7% measured from real videos. These results suggest that SPS-GAN can generate coherent motion from real-world video observations while reproducing aspects of the underlying energy behavior.

### 5.3 Identifying Active Degrees of Freedom

We next examine whether the learned latent phase space captures the effective degrees of freedom of the observed dynamics. The effective degrees of freedom characterize the number of independent coordinates needed to describe the observed motion. Identifying this dimensionality from trajectory or video observations provides insight into the underlying mechanical structure without requiring prescribed generalized coordinates. SPS-GAN encourages compact latent representations through

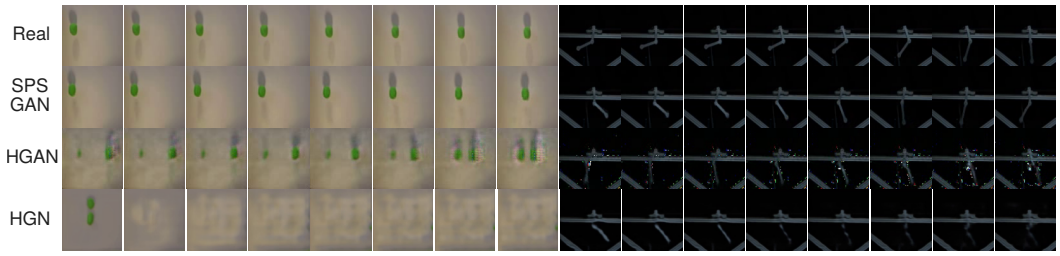



Figure 2: **Video generation on real-world pendulum systems.** Qualitative comparison of SPS-GAN, HGN, and HGAN on **(left)** single-pendulum [41] and **(right)** double-pendulum [42] videos. SPS-GAN generates temporally coherent motion, whereas the baselines exhibit static or dynamically inconsistent sequences. For the single pendulum, the relative energy variance of centroid-extracted SPS-GAN videos (2.9%) is close to that of real recordings (2.7%).

Table 5: **Identifying active degrees of freedom.** We report the percentage variance explained (PVE) by the leading principal components of the learned phase space. Components contributing to 95% cumulative PVE are highlighted in green. The estimated active dimensionality matches the expected effective degrees of freedom across simulated and real-world trajectory and video datasets.

| | Mass Spring | Sin. Pend. | Dou. Pend. | Two Body | Three Body | Damp. Pend. | Force. Duff. | Sin. Pend. | Dou. Pend. | Sin. Pend. | Dou. Pend. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mode | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Vid. | Vid. |
| Source | Sim | Sim | Sim | Sim | Sim | Sim | Sim | Real | Real | Real | Real |
| $\text{PVE}_1$ | 98.85 | 97.16 | 78.70 | 95.87 | 81.87 | 95.79 | 96.47 | 96.28 | 61.39 | 98.69 | 70.47 |
| $\text{PVE}_2$ | 0.48 | 0.79 | 18.39 | 1.32 | 17.40 | 1.28 | 1.02 | 1.91 | 35.41 | 0.64 | 24.63 |
| $\text{PVE}_3$ | 0.18 | 0.58 | 0.90 | 1.05 | 0.24 | 0.79 | 0.76 | 0.70 | 1.96 | 0.46 | 0.88 |
| DoF | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 |

cyclic-coordinate regularization, and we evaluate their active dimensionality using principal component analysis (PCA). Specifically, we perform PCA [43, 44] on the learned coordinate trajectories $\mathbf{q}_{0:T}$ from 4,096 randomly sampled phase-space rollouts. We compute the percentage variance explained (PVE) by principal component $k$ as

$$\text{PVE}_k = \lambda_k / \sum_i \lambda_i, \tag{14}$$

where $\lambda_k$ is the corresponding eigenvalue of the covariance matrix. We estimate the active dimension as the smallest number of components whose cumulative PVE exceeds 95% [45]. Full details and 10-component PVE results are provided in Appendix B.5.

We use $d_{\text{lat}} = 10$ across all experiments, providing an overparameterized latent phase space relative to the expected effective degrees of freedom. We evaluate five conservative and two non-conservative simulated systems, two real-world Cartesian trajectory datasets [42, 46], and two real-world video datasets [41, 42]. As shown in Table 5, the number of principal components required to explain 95% of the variance matches the expected effective degrees of freedom in all 11 settings. The learned representations exhibit one active dimension for the mass-spring, single-pendulum, and two-body systems, and two for the double-pendulum and three-body systems. The expected dimensions are also identified from real-world trajectory and video observations, suggesting that these findings extend beyond idealized simulated data. Together, these results indicate that the dominant directions of variation in SPS-GAN's learned latent coordinates reflect the effective dimensionality of the observed dynamics. Analytical derivations of the effective degrees of freedom are provided in Appendix B.4.

### 5.4 Identifying Interpretable Mechanical Coordinates

Identifying the active latent dimensions establishes the effective dimensionality of the learned representation, but does not reveal the physical quantities those dimensions encode. We therefore examine whether the active latent directions correspond to known mechanical coordinates by varying them individually and decoding the resulting states into observation space.

We use the PCA decomposition from Section 5.3 to identify the active principal directions $\mathbf{a}_k$ of the learned generalized coordinates. To interpret each direction, we vary its coordinate while holding the remaining principal components and latent momenta fixed. Specifically, we sample five states along

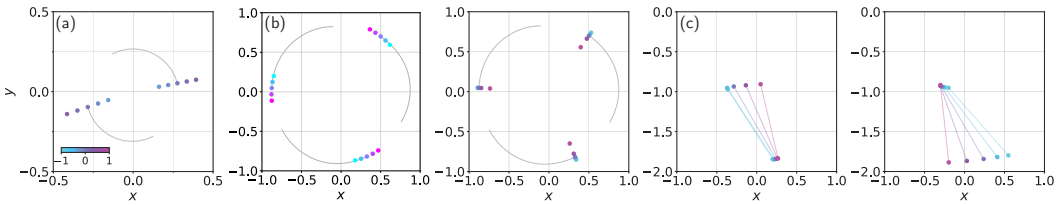



Figure 3: **Interpretable coordinates from active latent directions.** Sweeping active latent directions and decoding to Cartesian space reveals coordinates aligned with known mechanical variables: **(a)** radial distance for two-body; **(b)** orbital angle and radial scale for three-body; **(c)** two angular positions for double pendulum.

Table 6: **Ablation analysis on two-body system.** We ablate the phase-space map, cyclic-coordinate loss, and pHNN module. Each ablation degrades both accuracy and recovered active dimension.

| | SPS-GAN | +SympNet | +GRU | w/o Phase Space Map | w/o Cyclic Loss |
|---|---|---|---|---|---|
| MSE | $\mathbf{3.07\times10^{-3}}$ | $4.43 \times 10^{-3}$ | $1.60 \times 10^{-1}$ | $4.93 \times 10^{-2}$ | $4.40 \times 10^{-2}$ |
| DoF | 1 | 2 | 3 | 7 | 3 |

each direction and decode them into Cartesian space,

$$\mathbf{q}_k^{(i)} = \mathbf{q}^{(0)} + i\sqrt{\lambda_k}\mathbf{a}_k, \qquad \mathbf{z}_k^{(i)} = (\mathbf{q}_k^{(i)}, \mathbf{p}^{(0)}), \qquad \mathbf{x}_k^{(i)} = G_{\mathrm{T}}\left(\mathbf{z}_k^{(i)}\right), \tag{15}$$

$\lambda_k$ is the corresponding PCA eigenvalue, and $(\mathbf{q}^{(0)}, \mathbf{p}^{(0)})$ is a reference latent state. The decoded sweeps allow us to examine how each active latent direction relates to physical motion.

Figure 3 shows decoded sweeps for three representative systems. For the two-body system, the active direction corresponds to changes in radial separation $r$. For the three-body system, the two active directions correspond to radial scale and orbital angle, while those of the double pendulum correspond to the two pendulum angles $\theta_1$ and $\theta_2$. Additional results for the mass-spring, single pendulum, and single-pendulum video are provided in Appendix B.6. These results demonstrate that the learned active directions can correspond to physically interpretable mechanical coordinates, despite training only on Cartesian trajectories or video observations.

### 5.5 Ablation Analysis

We evaluate the contributions of the phase-space map, cyclic-coordinate regularization, and structured latent dynamics to trajectory accuracy and active dimensionality. On the two-body system, we remove the phase-space map or cyclic-coordinate loss, and replace the pHNN dynamics with either SympNet [47] or GRU [48]. As shown in Table 6, the complete SPS-GAN achieves the lowest trajectory MSE and recovers the expected single active dimension. Replacing the pHNN with SympNet or GRU increases trajectory error and the estimated active dimension, with the GRU exhibiting substantially larger errors. Removing the phase-space map produces the largest dimensionality discrepancy, increasing the estimated active dimension from one to seven, while removing cyclic-coordinate regularization increases it to three. These results indicate that the initialization map, structured dynamics, and cyclic-coordinate regularization make complementary contributions to dynamical fidelity and compact latent representations.

Additional ablations and robustness analyses are provided in Appendix C. These examine sensitivity to latent dimensionality, PCA thresholds, observation noise, training progress, and regularization weights. We also evaluate dimensionality estimates using eigenvalue-ratio and inactive-component pruning diagnostics, and assess the coverage of generated initial conditions.

## 6 Conclusion

We introduced SPS-GAN, a generative framework that combines structured latent dynamics with adversarial learning to generate trajectories and videos while discovering interpretable mechanical structure. By integrating port-Hamiltonian dynamics with cyclic-coordinate regularization, SPS-GAN models conservative, dissipative, and forced systems across trajectory and video observations. Experiments on simulated and real-world systems demonstrate strong dynamical fidelity and video-generation quality relative to established baselines. The results suggest that generative models equipped with structured latent dynamics can discover compact, physically interpretable representations directly from trajectory and video observations.

**Limitations.** SPS-GAN assumes sufficiently smooth dynamics that admit an approximate port-Hamiltonian representation, limiting its applicability to strongly stochastic or discontinuous systems. Active dimensionality is estimated from a finite-sample PCA spectrum and is sensitive to the variance threshold and observation noise. While dimensionality estimates remain consistent for thresholds of 90-95% and signal-to-noise ratios down to 10 dB in our experiments, stricter thresholds or greater noise can lead to overestimation. Finally, real-world evaluation is limited to low-dimensional trajectory recordings and planar video observations, leaving fully three-dimensional and contact-rich dynamics for future investigation.

## Acknowledgments and Disclosure of Funding

Yulong Yang was supported by a grant from the National Aeronautics and Space Administration funded through the Jet Propulsion Laboratory Strategic University Research Partnerships program. We sincerely thank the NeurIPS 2026 reviewers and area chair for their thoughtful feedback. The authors declare no competing interests.

## References

[1] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice Hall Upper Saddle River, NJ, 2002.

[2] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1, 2014.

[3] Clara O'Farrell and Ian G Clark. Reconstructed performance of mars 2020 parachute decelerator system. *Journal of Spacecraft and Rockets*, 61(5):1313–1328, 2024.

[4] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[5] Nicholas Galioto and Alex Arkady Gorodetsky. Bayesian system id: optimal management of parameter, model, and measurement uncertainty. *Nonlinear Dynamics*, 102(1):241–267, 2020.

[6] Juan Paredes and Dennis S Bernstein. Identification of self-excited systems using discrete-time, time-delayed lur'e models. In *2021 American Control Conference (ACC)*, pages 3939–3944. IEEE, 2021.

[7] Fan Wu, Woojin Cho, David Korotky, Sanghyun Hong, Donsub Rim, Noseong Park, and Kookjin Lee. Identifying contemporaneous and lagged dependence structures by promoting sparsity in continuous-time neural networks. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2534–2543, 2024.

[8] Juan A Paredes, Yulong Yang, and Dennis S Bernstein. Output-only identification of self-excited systems using discrete-time lur'e models with application to a gas-turbine combustor. *International Journal of Control*, 97(2):187–212, 2024.

[9] Riley J Richards, Yulong Yang, Juan A Paredes, and Dennis S Bernstein. Output-only identification of lur'e systems with hysteretic feedback nonlinearities. In *2024 American Control Conference (ACC)*, pages 2891–2896. IEEE, 2024.

[10] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

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

[12] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

[13] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Dissipative symoden: Encoding hamiltonian dynamics with dissipation and control into deep learning. *arXiv preprint arXiv:2002.08860*, 2020.

[14] Andrew Sosanya and Sam Greydanus. Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately. *arXiv preprint arXiv:2201.10085*, 2022.

[15] Shaan A Desai, Marios Mattheakis, David Sondak, Pavlos Protopapas, and Stephen J Roberts. Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E*, 104(3):034312, 2021.

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

[17] Christine Allen-Blanchette. Hamiltonian gan. In *6th Annual Learning for Dynamics & Control Conference*, pages 1662–1674. PMLR, 2024.

[18] Muhammad Firmansyah Kasim and Yi Heng Lim. Constants of motion network. *Advances in Neural Information Processing Systems*, 35:25295–25305, 2022.

[19] Takashi Matsubara and Takaharu Yaguchi. Finde: Neural differential equations for finding and preserving invariant quantities. *arXiv preprint arXiv:2210.00272*, 2022.

[20] Yana Lishkova, Paul Scherer, Steffen Ridderbusch, Mateja Jamnik, Pietro Liò, Sina Ober-Blöbaum, and Christian Offen. Discrete lagrangian neural networks with automatic symmetry discovery. *IFAC-PapersOnLine*, 56(2):3203–3210, 2023.

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

[22] Renyi Chen and Molei Tao. Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps. In *International conference on machine learning*, pages 1717–1727. PMLR, 2021.

[23] Marios Mattheakis, David Sondak, Akshunna S Dogra, and Pavlos Protopapas. Hamiltonian neural networks for solving equations of motion. *Physical Review E*, 105(6):065305, 2022.

[24] Amine Mohamed Aboussalah and Abdessalam Ed-dib. Geohnns: Geometric hamiltonian neural networks. *arXiv preprint arXiv:2507.15678*, 2025.

[25] 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.

[26] Justice J Mason, Christine Allen-Blanchette, Nicholas Zolman, Elizabeth Davison, and Naomi Ehrich Leonard. Learning to predict 3d rotational dynamics from images of a rigid body with unknown mass distribution. *Aerospace*, 10(11):921, 2023.

[27] 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.

[28] Asif Khan and Amos J Storkey. Hamiltonian latent operators for content and motion disentanglement in image sequences. *Advances in Neural Information Processing Systems*, 35:7250–7263, 2022.

[29] Haoran Li, Chenhan Xiao, Muhao Guo, and Yang Weng. Latent mixture of symmetries for sample-efficient dynamic learning. *arXiv preprint arXiv:2510.03578*, 2025.

[30] Jerry B Marion. *Classical dynamics of particles and systems*. Academic Press, 2013.

[31] Bernhard M Maschke and Arjan J van der Schaft. Port-controlled hamiltonian systems: modelling origins and systemtheoretic properties. In *Nonlinear control systems design 1992*, pages 359–365. Elsevier, 1993.

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

[33] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[34] In Huh, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Time-reversal symmetric ode network. *Advances in Neural Information Processing Systems*, 33:19016–19027, 2020.

[35] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[36] Ilan Naiman, N Benjamin Erichson, Pu Ren, Michael W Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman vaes. In *International conference on learning representations*, volume 2024, pages 50547–50573, 2024.

[37] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, volume 2024, pages 41582–41610, 2024.

[38] Manuel Brenner, Florian Hess, Jonas M Mikhaeil, Leonard F Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable dendritic rnns for reconstructing nonlinear dynamical systems. In *International conference on machine learning*, pages 2292–2320. Pmlr, 2022.

[39] Fan Wu, Cheng Jing, and Kookjin Lee. $F^3$no: Frequency-decomposed finite-time flow-map neural operators with cross-scale conditioning. *arXiv preprint arXiv:2610.10998*, 2026.

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

[41] Alejandro Castañeda Garcia, Jan Warchocki, Jan van Gemert, Daan Brinks, and Nergis Tomen. Learning physics from video: Unsupervised physical parameter estimation for continuous dynamical systems. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27924–27933, 2025.

[42] Kadierdan Kaheman, Urban Fasel, Jason J Bramburger, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. The experimental multi-arm pendulum on a cart: A benchmark system for chaos, learning, and control. *HardwareX*, 15:e00465, 2023.

[43] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[44] Ian Jolliffe. *Principal Component Analysis*. John Wiley & Sons, Ltd, 2014.

[45] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning: with applications in R*, volume 2. Springer, 2021.

[46] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[47] Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, and George Em Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

[48] Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1724–1734, 2014.

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

[50] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[52] Yulong Yang, Felix O'Mahony, and Christine Allen-Blanchette. Learning color equivariant representations. *arXiv preprint arXiv:2406.09588*, 2024.

[53] Tao Zhong and Christine Allen-Blanchette. Gagrasp: Geometric algebra diffusion for dexterous grasping. *arXiv preprint arXiv:2503.04123*, 2025.

[54] Yulong Yang, Bowen Feng, Keqin Wang, Naomi Ehrich Leonard, Adji Bousso Dieng, and Christine Allen-Blanchette. Behavior-inspired neural networks for relational inference. *arXiv preprint arXiv:2406.14746*, 2024.

[55] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

# A Dataset and Training

This appendix documents the datasets and training details used in Section 5, covering both simulated and real-world systems. The simulated data comes from two sources: five conservative systems (mass-spring oscillator, ideal pendulum, double pendulum, two-body, and three-body) and two non-conservative systems (damped pendulum and forced Duffing oscillator) generated using the HGN re-implementation pipeline [49].

## A.1 Dataset Generation

All simulated trajectories and video are produced via integration with timestep $\Delta t = 0.05$ and length $T = 30$ frames. For the long-horizon rollout example in Table 2, the double pendulum is simulated to $T = 200$ frames. The real-world data comprises pendulum trajectory recordings [46], double pendulum trajectory recordings [42], and double pendulum video recordings [42]. We detail the equations of motion, initial conditions, and sampling procedures for each system presented in Section 5.

### A.1.1 Mass-Spring System.

The mass-spring oscillator is governed by the Hamiltonian

$$\mathcal{H} = \frac{p^2}{2m} + \frac{1}{2}ky^2, \tag{16}$$

with dynamics

$$\ddot{y} = -2c\sqrt{k/m}\dot{y} - (k/m)y. \tag{17}$$

Our experiments use mass $m = 0.5$ and elastic constant $k = 2.0$. The initial position $y_0$ is sampled uniformly from $[0.1, 1.0]$. As the mass-spring is a one-dimensional system, we pad with $x = 0.0$ to embed the trajectory in the 2D Cartesian frame used by the other systems.

### A.1.2 Duffing Oscillator.

The conservative Duffing oscillator is governed by the Hamiltonian

$$\mathcal{H} = \frac{1}{2}p^2 + \frac{1}{2}\alpha q^2 + \frac{1}{4}\beta q^4, \tag{18}$$

with dynamics of the forced case being

$$\ddot{x} = \gamma \cos \omega t - \delta \dot{x} - \alpha x - \beta x^3. \tag{19}$$

Our experiments use forcing $\delta = 0.15$, $\alpha = -0.2$, and $\beta = 0.2$. The initial position $x_0$ is sampled uniformly from $[0.2, 1.0]$. As the Duffing oscillator is a one-dimensional system, we pad with $y = 0.0$ to embed the trajectory in the 2D Cartesian frame used by the other systems.

### A.1.3 Ideal Pendulum.

The ideal pendulum system is governed by the Hamiltonian

$$\mathcal{H} = \frac{p^2}{2mL^2} + mgL(1 - \cos\theta), \tag{20}$$

with dynamics

$$\ddot{\theta} = -(g/L)\sin\theta. \tag{21}$$

Our experiments use mass $m = 0.5$, length $L = 1.0$, and gravity $g = 9.81$. The initial angular position is sampled uniformly from $[-\pi, \pi]$. While the system is simulated using canonical coordinates, the final training trajectory is projected to Cartesian coordinates or video sequences.

### A.1.4 Double Pendulum.

The chaotic double pendulum system is represented by the Hamiltonian

$$\mathcal{H} = \frac{1}{2mL^2} \cdot \frac{p_1^2 + 2p_2^2 - 2p_1 p_2 \cos(\theta_1 - \theta_2)}{1 + \sin^2(\theta_1 - \theta_2)} + mgL(3 - 2\cos\theta_1 - \cos\theta_2), \tag{22}$$

where $\theta_1, \theta_2$ denote the angular positions and $p_1, p_2$ denote the corresponding conjugate momenta. The dynamics are given by Hamilton's equations. Let $\Delta = \theta_1 - \theta_2$, $D = 1 + \sin^2(\Delta)$, and $N = p_1^2 + 2p_2^2 - 2p_1 p_2 \cos(\Delta)$. Then,

$$\dot{\theta}_1 = \frac{p_1 - p_2 \cos(\Delta)}{mL^2 D}, \tag{23}$$

$$\dot{\theta}_2 = \frac{2p_2 - p_1 \cos(\Delta)}{mL^2 D}, \tag{24}$$

$$\dot{p}_1 = \frac{\sin(\Delta)(N\cos(\Delta) - p_1 p_2 D)}{mL^2 D^2} - 2mgL\sin\theta_1, \tag{25}$$

$$\dot{p}_2 = \frac{\sin(\Delta)(p_1 p_2 D - N\cos(\Delta))}{mL^2 D^2} - mgL\sin\theta_2, \tag{26}$$

with parameters $m = 1.0$, $L = 1.0$, and $g = 3.0$. The initial angular position is sampled uniformly from $[-0.5\pi, 0.5\pi]$. While the system is simulated using canonical coordinates, the final training trajectory is projected to Cartesian coordinates or video sequences.

### A.1.5 Planar Two-Body System.

The gravitational two-body system is represented by the Hamiltonian

$$\mathcal{H} = \frac{|\mathbf{p}_1|^2}{2m_1} + \frac{|\mathbf{p}_2|^2}{2m_2} - \frac{Gm_1 m_2}{|\mathbf{q}_1 - \mathbf{q}_2|}, \tag{27}$$

where the dynamics are governed by

$$m_i \ddot{\mathbf{q}}_i = G \sum_{j \neq i} \frac{m_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{|\mathbf{q}_j - \mathbf{q}_i|^3}, \tag{28}$$

with equal masses $m = 1.0$ and gravitational constant $G = 1.0$. The two objects are placed on the same circular orbit at opposite position with the radius ranging from 0.5 to 1.5. Their initial velocities are set perpendicular to the radius vector, and a small perturbations of magnitude $0.1$ is added to the velocities.

### A.1.6 Planar Three-Body System.

The gravitational three-body system is represented by the Hamiltonian

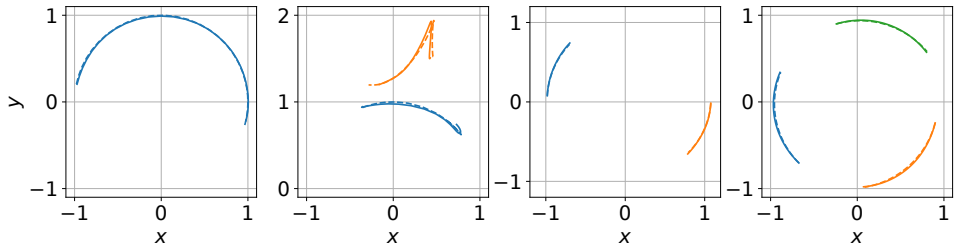
$$\mathcal{H} = \sum_{i=1}^{3} \frac{|\mathbf{p}_i|^2}{2m_i} - \sum_{i<j} \frac{Gm_i m_j}{|\mathbf{q}_i - \mathbf{q}_j|}, \tag{29}$$

with equal masses $m = 1.0$ and gravitational constant $G = 1.0$. Three equal mass objects are initialized in an equilateral triangle configuration with $120^\circ$ angular separation within radius bounds $[0.9, 1.5]$. Their initial velocities are set perpendicular to the radius vector, and a small perturbations of magnitude $0.1$ is added to the velocities.

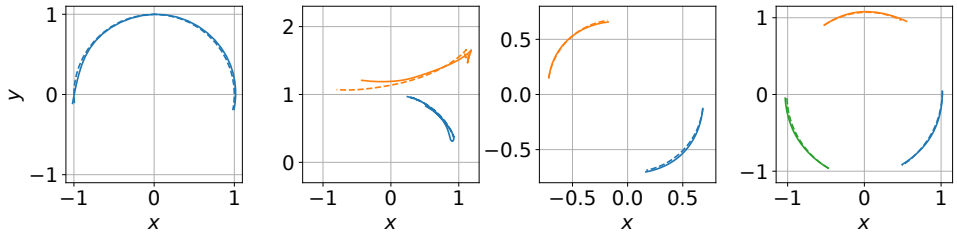
## A.2 Training Setting

All training of SPS-GAN was performed on cluster nodes each with 1 Nvidia L40 GPU, 16 core partition of an Intel Xeon Gold 5320, and 64G of DDR4 3200MHz RDIMM.

**SPS-GAN trajectory training hyperparameter.** SPS-GAN for generating time-series data is trained through 20000 steps with batch size of 128 (pendulum, double pendulum, mass-spring, two-body) or 160 (three-body). It is optimized using Adam [50] with $\beta_1 = 0.3$ and $\beta_2 = 0.999$ using a learning rate of $5 \times 10^{-5}$. The phase-space map and HNN module is initialized as an MLP with hidden size



(a) **SPS-GAN-1 unsupervised modeling of single systems.**



(b) **SPS-GAN-5 joint modeling of five systems.**

Figure 4: **Example Cartesian trajectory rollout.** Comparison between ground-truth (dotted) and generated (solid) trajectories across the single pendulum, double pendulum, two-body, and three-body systems. Whether generating dynamics from **(a)** one system at a time or **(b)** multiple systems jointly, SPS-GAN trajectories closely follow the reference dynamics.

100 using ReLU activation. The trajectory generator is initialized as an MLP with hidden size 512 using SoftPlus activation. We set the latent motion space to $d_{\text{m}} = 10$, content space to $d_{\text{c}} = 0$, and output size $d_{\text{output}} = 10$. For training we use trajectories of length 30; for discrimination we set $\ell = 16$.

**SPS-GAN video training hyperparameter.** SPS-GAN for generating video data is trained through 20000 steps with batch size of 16. It is optimized using Adam with $\beta_1 = 0.3$ and $\beta_2 = 0.999$ using a learning rate of $5 \times 10^{-5}$. The phase-space map and pHNN module is initialized as an MLP with hidden size 100 using ReLu activation. The image generator is initialized as an CNN with 32 filter and 3 channel. We set the latent motion space to $d_{\text{m}} = 10$, content space to $d_{\text{c}} = 50$. For training we use video sequences of length 30; for discrimination we set $\ell = 16$.

# B Results

We expand on the analysis in Section 5 with example trajectory and video rollouts, extended ablation analysis, robustness of SPS-GAN to the choice of latent dimension $d_{\text{lat}}$, and analytical derivations of the degrees of freedom and canonical coordinates for each system.

## B.1 Example Cartesian Trajectory Rollout

We include example Cartesian trajectory rollouts generated by SPS-GAN in Figure 4 for two training regimes: SPS-GAN-1, trained without supervision on one system at a time, and SPS-GAN-5, trained jointly on all five systems with supervisory conditional labels.

For systems reported in Table 3, the learnable dissipation $\mathbf{D}_\phi$ parameter and input map $\mathbf{F}_\psi$ accommodates the non-conservative dynamics in Equation (3). We constraint the dissipation term $\mathbf{D}_\phi$ to be positive semidefinite through Cholesky decomposition such that $\mathbf{D}_\phi = \tilde{\mathbf{D}}_\phi \tilde{\mathbf{D}}_\phi^\top$.

## B.2 Example Video Rollout

We include example video rollouts generated by SPS-GAN in Figure 5, trained without supervision on one system at a time with no color or physics variation. We include example video rollouts generated by SPS-GAN-5 jointly modeling all five systems in Figure 6, trained with conditional labels encoding variations in color and physics.

### B.3 Controlled Pendulum

We include additional results on the generated trajectory of unsupervised SPS-GAN on a controlled pendulum, using the Gymnasium environment [51]. At test time SPS-GAN generates a trajectory under a constant input. The first coordinate of the generated trajectory and input are used to create the ground truth reference using the gymnasium environment. We show that SPS-GAN is able to accurate learned the dynamics under external actuation in Figure 7, where the generated trajectory closely matches the gymnasium simulated one. Notably the SPS-GAN generated trajectory deviates significantly from the unactuated dynamics, indicating that SPS-GAN is able to learn the dynamics of actuation rather than small perturbations. Empirically, SPS-GAN generated controlled pendulum yielded MSE of $4.91 \times 10^{-4}$, on par with the accuracy of unactuated ideal pendulum (see Table 1).

### B.4 Analytical Derivation

We show the analytical derivation of the degrees of freedom and canonical coordinates for the systems studied in Section 5, and verify that the identified degrees of freedom and recovered canonical coordinates match the analytical results.

**Mass Spring.** The mass-spring oscillator is described by a single generalized coordinate $y$ (the displacement from equilibrium), giving one degree of freedom by definition.

**Duffing Oscillator.** The Duffing oscillator is described by a single generalized coordinate $x$ (the displacement), giving one degree of freedom by definition.

**Single Pendulum.** The Cartesian coordinate $(x, y)$ can be parameterized by the angular position $\theta$

$$x = L\sin\theta, \qquad y = -L\cos\theta, \tag{30}$$

where $L$ is the length of the rod. The kinetic and potential energies, along with damping are

$$T = \frac{1}{2}mL^2\dot{\theta}^2, \tag{31}$$

$$V = -mgL\cos\theta, \tag{32}$$

$$\mathcal{D} = \frac{1}{2}c\dot{\theta}^2, \tag{33}$$

where $c$ is the damping coefficient. Therefore the unactuated Euler-Lagrange equation of Lagrangian $\mathcal{L} = T - V$ with damping becomes

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}}\right) - \frac{\partial\mathcal{L}}{\partial\theta} + \frac{\partial\mathcal{D}}{\partial\dot{\theta}} = 0. \tag{34}$$

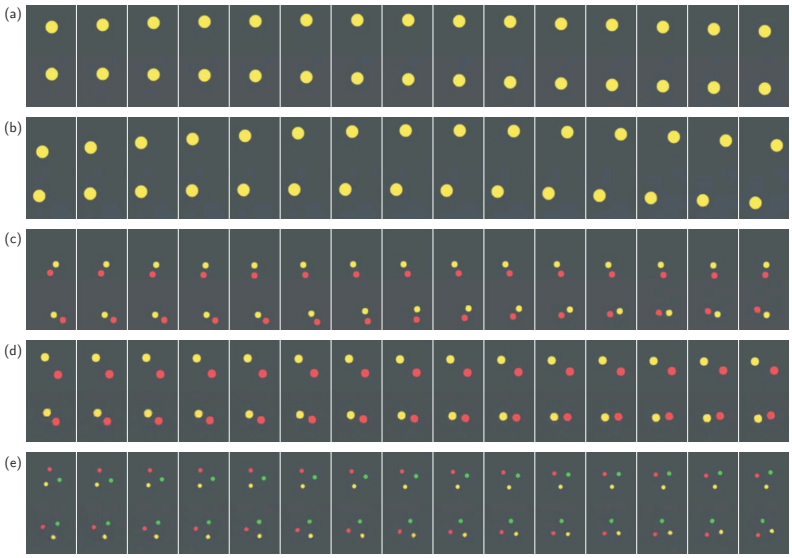



Figure 5: **Example video rollout under fixed parameters.** Real system (top) and generated video (bottom) for (a) mass-spring oscillator, (b) ideal pendulum, (c) double pendulum, (d) two-body, and (e) three-body system, all under fixed parameters.

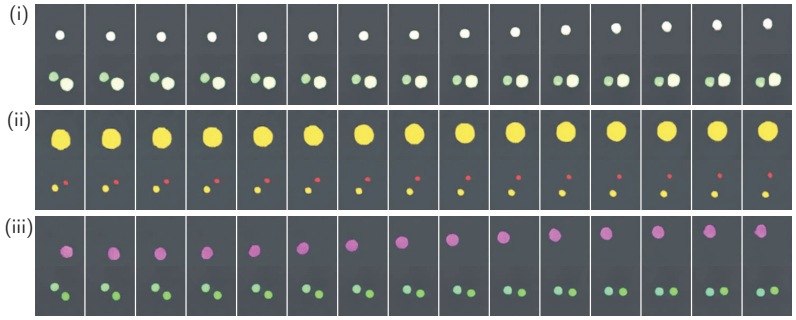



Figure 6: **Generated video under color and physics variation.** SPS-GAN-5 generated video of mass-spring and two body systems with **(i)** physics and color varied (FVD 182.15), **(ii)** only physics varied (FVD 135.38), and **(iii)** only color varied (FVD 135.25). The default color setting uses yellow, red, and green.

Solving and expanding the terms yields the dynamics equation for the single pendulum

$$\ddot{\theta} = -\frac{c}{mL^2}\dot{\theta} - \frac{g}{L}\sin\theta, \tag{35}$$

which only dependents on canonical coordinate $\theta$, showing a one degree of freedom system.

**Double Pendulum.** The Cartesian coordinates of the two masses can be parameterized by the angular positions $\theta_1$ and $\theta_2$ as

$$x_1 = L_1 \sin\theta_1, \qquad y_1 = -L_1\cos\theta_1, \tag{36}$$
$$x_2 = L_1\sin\theta_1 + L_2\sin\theta_2, \qquad y_2 = -L_1\cos\theta_1 - L_2\cos\theta_2, \tag{37}$$

where $L_1$ and $L_2$ are the lengths of the two rods. The angles are measured from the downward vertical, with $\theta_1$ describing the first rod and $\theta_2$ describing the second rod. Let

$$\Delta = \theta_1 - \theta_2. \tag{38}$$

Using the coordinate parameterization above, the squared speeds of the two masses are

$$v_1^2 = L_1^2\dot{\theta}_1^2, \tag{39}$$
$$v_2^2 = L_1^2\dot{\theta}_1^2 + L_2^2\dot{\theta}_2^2 + 2L_1L_2\cos\Delta\,\dot{\theta}_1\dot{\theta}_2. \tag{40}$$

The kinetic and potential energies, along with damping, are therefore

$$T = \frac{1}{2}(m_1+m_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2L_2^2\dot{\theta}_2^2 + m_2L_1L_2\cos\Delta\,\dot{\theta}_1\dot{\theta}_2, \tag{41}$$
$$V = -(m_1+m_2)gL_1\cos\theta_1 - m_2gL_2\cos\theta_2, \tag{42}$$
$$\mathcal{D} = \frac{1}{2}c_1\dot{\theta}_1^2 + \frac{1}{2}c_2(\dot{\theta}_2 - \dot{\theta}_1)^2, \tag{43}$$

where $c_1$ is the damping coefficient at the first pivot, and $c_2$ is the damping coefficient at the second joint. The second damping term depends on the relative angular velocity between the two rods. The

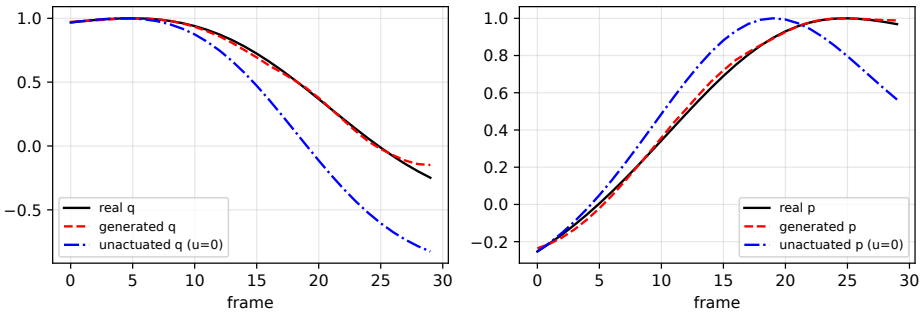



Figure 7: **Accuracy of generated trajectory for actuated pendulum.** SPS-GAN generated trajectory under constant external actuation closely matches the reference dynamics from gymnasium[51].

Lagrangian is $\mathcal{L} = T - V$, and the unactuated Euler-Lagrange equations with damping are

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} + \frac{\partial \mathcal{D}}{\partial \dot{\theta}_i} = 0, \quad i = 1, 2. \tag{44}$$

For compactness, define

$$A = (m_1 + m_2)L_1^2, \qquad B = m_2 L_1 L_2, \qquad C = m_2 L_2^2, \tag{45}$$

$$G_1 = (m_1 + m_2)gL_1, \qquad G_2 = m_2 g L_2. \tag{46}$$

Solving and expanding the terms yields the coupled dynamics of the damped double pendulum,

$$A\ddot{\theta}_1 + B\cos\Delta\,\ddot{\theta}_2 + B\sin\Delta\,\dot{\theta}_2^2 + G_1\sin\theta_1 + (c_1 + c_2)\dot{\theta}_1 - c_2\dot{\theta}_2 = 0, \tag{47}$$

$$B\cos\Delta\,\ddot{\theta}_1 + C\ddot{\theta}_2 - B\sin\Delta\,\dot{\theta}_1^2 + G_2\sin\theta_2 - c_2\dot{\theta}_1 + c_2\dot{\theta}_2 = 0. \tag{48}$$

Equivalently, the system can be written explicitly in terms of the angular accelerations. Define

$$R_1 = -B\sin\Delta\,\dot{\theta}_2^2 - G_1\sin\theta_1 - (c_1 + c_2)\dot{\theta}_1 + c_2\dot{\theta}_2, \tag{49}$$

$$R_2 = B\sin\Delta\,\dot{\theta}_1^2 - G_2\sin\theta_2 + c_2\dot{\theta}_1 - c_2\dot{\theta}_2. \tag{50}$$

Then the equations of motion are

$$\ddot{\theta}_1 = \frac{CR_1 - B\cos\Delta\,R_2}{AC - B^2\cos^2\Delta}, \tag{51}$$

$$\ddot{\theta}_2 = \frac{-B\cos\Delta\,R_1 + AR_2}{AC - B^2\cos^2\Delta}. \tag{52}$$

Thus, the damped double pendulum depends on the two canonical coordinates $\theta_1$ and $\theta_2$, showing a two-degree-of-freedom system.

**Planar Two-Body System.** Consider two point masses $m_1$ and $m_2$ moving in a plane under their mutual gravitational attraction. Let $(x_1, y_1)$ and $(x_2, y_2)$ denote the Cartesian positions of the two bodies, and define the relative coordinate from body 1 to body 2 as

$$x = x_2 - x_1, \qquad y = y_2 - y_1. \tag{53}$$

The relative coordinate can be parameterized in polar form by the radius $r$ between the bodies and the angle $\theta$ as

$$x = r\cos\theta, \qquad y = r\sin\theta, \qquad r = \sqrt{x^2 + y^2}. \tag{54}$$

Let $M = m_1 + m_2$ be the total mass and let $\mu = \frac{m_1 m_2}{m_1 + m_2}$ be the reduced mass. After separating out the center-of-mass motion, the kinetic and potential energies of the relative system are

$$T = \frac{1}{2}\mu\left(\dot{r}^2 + r^2\dot{\theta}^2\right), \tag{55}$$

$$V = -\frac{Gm_1 m_2}{r}, \tag{56}$$

where $G$ is the gravitational constant. Therefore the two-body problem can be written as an equivalent one-body problem with reduced mass $\mu$ moving in a central potential. The Lagrangian is

$$\mathcal{L} = T - V = \frac{1}{2}\mu\left(\dot{r}^2 + r^2\dot{\theta}^2\right) + \frac{Gm_1 m_2}{r}. \tag{57}$$

The unactuated Euler-Lagrange equations are

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0, \qquad q_i \in \{r, \theta\}. \tag{58}$$

Since the Lagrangian does not explicitly depend on $\theta$, the angular coordinate is cyclic, so its conjugate momentum is conserved

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = \mu r^2\dot{\theta} = \ell, \tag{59}$$

where $\ell$ is the conserved angular momentum of the reduced system. Equivalently, defining the specific angular momentum $h = \frac{\ell}{\mu}$ gives

$$r^2\dot{\theta} = h, \qquad \dot{\theta} = \frac{h}{r^2}. \tag{60}$$

The Euler-Lagrange equation for the radial coordinate is

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{r}}\right) - \frac{\partial \mathcal{L}}{\partial r} = 0. \tag{61}$$

Expanding the terms gives

$$\mu\ddot{r} - \mu r\dot{\theta}^2 + \frac{Gm_1m_2}{r^2} = 0. \tag{62}$$

Solving for $\ddot{r}$ yields

$$\ddot{r} = r\dot{\theta}^2 - \frac{G(m_1+m_2)}{r^2}. \tag{63}$$

Using conservation of angular momentum to eliminate $\dot{\theta}$, the radial equation of motion becomes

$$\ddot{r} = \frac{h^2}{r^3} - \frac{G(m_1+m_2)}{r^2}. \tag{64}$$

Thus, after removing the center-of-mass motion, the relative two-body system has two degrees of freedom $(r, \theta)$. Conservation of angular momentum renders $\theta$ cyclic and recoverable by quadrature from $\dot{\theta} = h/r^2$, reducing the dynamics to an effective one-degree-of-freedom system in the radial coordinate $r$.

**Planar Three-Body Problem.** We denote by $q_j \in \mathbb{R}^2$ the position vector of the $j$th particle of mass $m_j$, where $j = 1, 2, 3$. The collision set is defined as

$$\Delta_{ij} = \{q = (q_1, q_2, q_3) \in (\mathbb{R}^2)^3 : q_i = q_j\}, \qquad \Delta = \cup_{1\le i<j\le 3}\Delta_{ij}. \tag{65}$$

We consider $q = (q_1, q_2, q_3) \in (\mathbb{R}^2)^3 \setminus \Delta$ in the center-of-mass frame, so that $\sum_{j=1}^3 m_j q_j = 0$. Let $M = \mathrm{diag}(m_1, m_1, m_2, m_2, m_3, m_3)$. With the gravitational constant absorbed into the units, the Newtonian potential and equations of motion are

$$U(q) = \sum_{1\le i<j\le 3} \frac{m_i m_j}{\|q_i - q_j\|}, \qquad M\ddot{q} = \nabla U(q). \tag{66}$$

We restrict to equilateral homographic solutions. Identifying $\mathbb{R}^2$ with $\mathbb{C}$, write

$$q_j(t) = z(t)q_j^{(c)}, \qquad j = 1, 2, 3, \tag{67}$$

where $q^{(c)} = (q_1^{(c)}, q_2^{(c)}, q_3^{(c)})$ is a fixed equilateral configuration satisfying $\sum_{j=1}^3 m_j q_j^{(c)} = 0$, and $z(t) \in \mathbb{C}$ controls the global scale and rotation of the triangle. The corresponding homographic initial velocities satisfy

$$\dot{q}_j(0) = \dot{z}(0)q_j^{(c)}, \qquad j = 1, 2, 3. \tag{68}$$

For the equal-mass case, this corresponds to initializing the three bodies at the vertices of an equilateral triangle and giving them equal velocity magnitudes separated by $120^\circ$. For example, if $q_j^{(c)} = \rho e^{\mathrm{i}(\alpha_0 + 2\pi(j-1)/3)}$ and $\dot{z}(0) = \frac{v}{\rho}e^{\mathrm{i}\beta_0}$, then

$$\dot{q}_j(0) = ve^{\mathrm{i}(\alpha_0+\beta_0+2\pi(j-1)/3)}, \qquad j = 1, 2, 3, \tag{69}$$

so the velocity vectors have equal magnitude and are separated by $120^\circ$. Let $a_c = \|q_i^{(c)} - q_j^{(c)}\|$ denote the side length of the reference equilateral triangle. Since $q^{(c)}$ is equilateral and centered at the center of mass,

$$\nabla_{q_j} U(q^{(c)}) = \sum_{k\neq j} \frac{m_j m_k (q_k^{(c)} - q_j^{(c)})}{a_c^3} = -\frac{m_j m_{\mathrm{tot}}}{a_c^3} q_j^{(c)}, \qquad m_{\mathrm{tot}} = m_1 + m_2 + m_3. \tag{70}$$

Thus $q^{(c)}$ is a central configuration satisfying

$$\nabla U(q^{(c)}) = -\lambda M q^{(c)}, \qquad \lambda = \frac{m_{\mathrm{tot}}}{a_c^3}. \tag{71}$$

Using the homogeneity of the Newtonian potential, the gradient satisfies

$$\nabla U(zq^{(c)}) = \frac{z}{|z|^3}\nabla U(q^{(c)}). \tag{72}$$

This central-configuration condition is the structural reason the homographic (i.e., the shape is preserved, only rotation and scaling is applied) approach is consistent, as substituting $q(t) = z(t)q^{(c)}$ into Newton's equations produces a right-hand side proportional to $\nabla U(q^{(c)})$, and only when this gradient is parallel to $Mq^{(c)}$ does the system collapse to a scalar equation in $z(t)$. For a non-central reference configuration, the gradient would have components transverse to $Mq^{(c)}$, the homographic form would fail to be preserved in time, and the reduction would break down. Substituting $q(t) = z(t)q^{(c)}$ into Newton's equations gives

$$\ddot{z}(t)Mq^{(c)} = \frac{z(t)}{|z(t)|^3}\nabla U(q^{(c)}) = -\lambda\frac{z(t)}{|z(t)|^3}Mq^{(c)}. \tag{73}$$

Therefore, the three-body dynamics restricted to the equilateral homographic subspace reduce to the scalar complex equation

$$\ddot{z} = -\lambda\frac{z}{|z|^3}. \tag{74}$$

Let

$$z(t) = r(t)e^{\mathrm{i}\phi(t)}. \tag{75}$$

Here $r(t)$ controls the size of the equilateral triangle, while $\phi(t)$ controls its global rotation. In particular, the physical side length is

$$a(t) = a_c r(t). \tag{76}$$

Thus, even if the reference configuration or units are normalized so that $\lambda = 1$, the equilateral size is still allowed to vary through the dynamic variable $r(t)$. Expanding $\ddot{z}$ gives

$$\ddot{z} = \left(\ddot{r} - r\dot{\phi}^2\right)e^{\mathrm{i}\phi} + \left(r\ddot{\phi} + 2\dot{r}\dot{\phi}\right)\mathrm{i}e^{\mathrm{i}\phi}. \tag{77}$$

Equating real and imaginary parts yields

$$\ddot{r} - r\dot{\phi}^2 = -\frac{\lambda}{r^2}, \qquad r\ddot{\phi} + 2\dot{r}\dot{\phi} = 0. \tag{78}$$

Solving for the second derivatives gives the reduced equations of motion

$$\ddot{r} = r\dot{\phi}^2 - \frac{\lambda}{r^2}, \qquad \ddot{\phi} = -2\frac{\dot{r}\dot{\phi}}{r}. \tag{79}$$

If the reference configuration or units are normalized so that $\lambda = 1$, then

$$\ddot{r} = r\dot{\phi}^2 - \frac{1}{r^2}, \qquad \ddot{\phi} = -2\frac{\dot{r}\dot{\phi}}{r}. \tag{80}$$

Therefore, under the equilateral homographic constraint, the planar three-body system reduces to a two-degree-of-freedom system with generalized coordinates $r$ and $\phi$. The coordinate $r$ describes the changing size of the equilateral triangle, while $\phi$ describes its global rotation.

Table 7: **Identifying system degrees of freedom.** SPS-GAN correctly recovers the degrees of freedom for conservative and non-conservative systems, under observations of Cartesian trajectory and video recordings. PCA components contributing to the 95% are highlighted in green.

| | Mass Spring | Sin. Pend. | Dou. Pend. | Two Body | Three Body | Damp. Pend. | Force. Duff. | Sin. Pend. | Dou. Pend. | Sin. Pend. | Dou. Pend. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mode | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Traj. | Video | Video |
| Source | Sim | Sim | Sim | Sim | Sim | Sim | Sim | Real | Real | Real | Real |
| $\mathrm{PVE}_1$ | 98.85 | 97.16 | 78.70 | 95.87 | 81.87 | 95.79 | 96.47 | 96.28 | 61.39 | 98.69 | 70.47 |
| $\mathrm{PVE}_2$ | 0.48 | 0.79 | 18.39 | 1.32 | 17.40 | 1.28 | 1.02 | 1.91 | 35.41 | 0.64 | 24.63 |
| $\mathrm{PVE}_3$ | 0.18 | 0.58 | 0.90 | 1.05 | 0.24 | 0.79 | 0.76 | 0.70 | 1.96 | 0.46 | 0.88 |
| $\mathrm{PVE}_4$ | 0.14 | 0.39 | 0.84 | 0.73 | 0.14 | 0.72 | 0.44 | 0.39 | 0.67 | 0.16 | 0.86 |
| $\mathrm{PVE}_5$ | 0.11 | 0.34 | 0.59 | 0.52 | 0.12 | 0.50 | 0.35 | 0.22 | 0.30 | 0.12 | 0.67 |
| $\mathrm{PVE}_{6\text{-}10}$ | 0.24 | 0.74 | 0.58 | 0.51 | 0.23 | 0.92 | 0.96 | 0.50 | 0.27 | 0.09 | 1.71 |
| DoF | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 |

Table 8: **Alignment of phase space at different times.** We report the absolute cosine similarity between the first two eigenvectors of the phase space for the three-body system at $t = \{0, 15, 30\}$, where the phase space at different times are geometrically equivalent

| $\mathbf{a}_1$ | $t = 0$ | $t = 15$ | $t = 30$ |
|---|---|---|---|
| $t = 0$ | 1.000 | 0.9997 | 0.9988 |
| $t = 15$ | 0.9997 | 1.000 | 0.9997 |
| $t = 30$ | 0.9988 | 0.9997 | 1.000 |

| $\mathbf{a}_2$ | $t = 0$ | $t = 15$ | $t = 30$ |
|---|---|---|---|
| $t = 0$ | 1.000 | 0.9992 | 0.9968 |
| $t = 15$ | 0.9992 | 1.000 | 0.9993 |
| $t = 30$ | 0.9968 | 0.9993 | 1.000 |

### B.5 Identifying Degree of Freedom

We measure the resulting dimensionality by performing PCA [43] on the latent trajectories and computing the percentage variance explained by each eigenvalue $\lambda_k$,

$$\text{PVE}_k = \lambda_k / \sum \lambda_k. \tag{81}$$

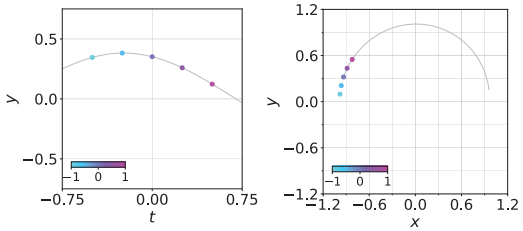
The system's degrees of freedom are then given by the smallest number of components whose cumulative PVE exceeds 95% [45]. We sample 4,096 random motion sample $\epsilon_m$ and encode them to phase space trajectories. We compute the PCA on these 4,096 phase space trajectories, yielding a 10-component PVE. We include the full 10-component PCA variance explained metric in Table 7.

We further try to understand if the structure of the latent phase space persists through time and initial conditions. Instead of performing PCA over the entire trajectory, we performance the PCA analysis on points corresponding to $t = \{0, 15, 30\}$. We report the absolute cosine similarity between the first two eigenvectors of the phase space of the three body system at $t = \{0, 15, 30\}$, where Table 8 shows that the phase space at different times are geometrically equivalent. Therefore the learned phase space is able to model the full dynamics with a single parameterization.

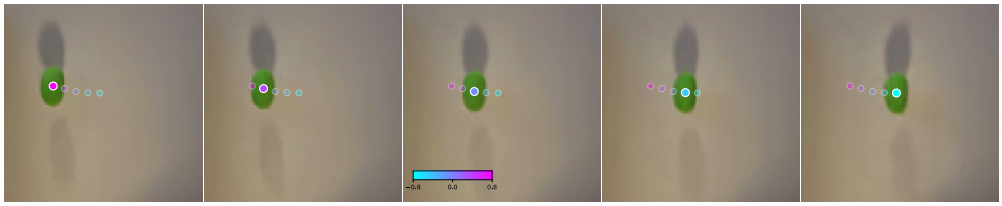
### B.6 Recovering Canonical Coordinates

We include the recovered canonical coordinates for mass spring and single pendulum in Figure 8a below. For the mass spring (left), sweeping the single active direction produces decoded positions that traces a sine wave in time, recovering the verticle displacement $y$ that defines the pendulum's motion. For the single pendulum (right), sweeping the single active direction produces decoded positions that trace an arc, recovering the angular coordinate $\theta$ that defines the pendulum's motion.

Furthermore, perform the same walk on the phase space and decode to video frames in Figure 8b, sweeping the single active direction produces decoded positions that trace an arc, recovering the




(a) **Recovering system canonical coordinates.** SPS-GAN recovers the correct canonical coordinates, specifically **(left)** displacement $y$ for mass spring and **(right)** angular position $\theta$ for single pendulum.



(b) **Recovering system canonical coordinates from video sequences.** SPS-GAN recovers the correct canonical coordinates $\theta$ from videos of a single pendulum [41].

Figure 8: **Recovering system canonical coordinates.** We show the recovered canonical coordinates for **(a)** simulated mass-spring and single pendulum, and **(b)** real world single pendulum videos.

Table 9: **Robustness of phase space dimension.** We train SPS-GAN with phase space with dimension $d_{\text{lat}} = \{10, 100, 1000\}$, where SPS-GAN is able to consistently recover a two degrees of freedom representation for the three-body system. Furthermore, increase phase space dimension yields marginal increase in computational cost, both in terms of memory and training time, suggesting that moderate overparameterization is practical. $\text{CPVE}_2$ denotes the cumulative PVE of the first two components.

| $d_{\text{lat}}$ | 10 | 100 | 1000 |
|---|---|---|---|
| $\lambda_{\text{cyclic}}$ | 0.03 | 0.03 | 0.03 |
| DoF | 2 | 2 | 2 |
| $\text{CPVE}_2$ | 99.27% | 99.16% | 99.20% |
| MSE | $1.26 \times 10^{-3}$ | $1.49 \times 10^{-3}$ | $1.75 \times 10^{-3}$ |
| ep/s | 0.236 | 0.233 | 0.221 |
| MB | 389 | 391 | 433 |

angular coordinate $\theta$ that defines the pendulum's motion. This demonstrates that the interpretability and phase-space structure extends between observation modalities.

# C Ablation and Robustness Analysis

## C.1 Robustness of Phase Space Dimension

**Latent-Space Upper Bound.** SPS-GAN does not assume prior knowledge on the size of the learned phase space. Therefore, in practice we choose $d_{\text{lat}}$ as an arbitrarily large upper bound on the expected degrees of freedom of the systems being modeled. We train SPS-GAN with $d_{\text{lat}} = \{10, 100, 1000\}$ in Table 9 on the three-body system. SPS-GAN is able to consistently recover the correct two degree of freedom representation with similar first two cumulative PVE (CPVE) and MSE. The exact dimension of the phase space also has minimal effects on the memory requirement and training time, suggesting that moderate overparameterization is computational practical.

**Sensitivity to the PVE Threshold.** We examine whether the recovered dimension depends on the choice of the 95% cumulative PVE threshold in Table 10. We vary the threshold from 90% to 99% across three training seeds (72, 172, 272) and report how many seeds recover the expected dimension in each setting. As shown in Table 10, both 90% and 95% recover the expected dimension for all three seeds across five simulated trajectory settings and two real video settings. Stricter thresholds increasingly count small residual-variance components as active, particularly for the two-body system and double-pendulum video. Recovery is therefore consistent over the tested 90–95% range, but is not invariant to the threshold.

**Eigenvalue-Ratio Diagnostic.** We further check whether the recovered dimension corresponds to a clear separation in the PCA spectrum. Using the same eigenvalues, we identify the index

$$\hat{d}_{\text{ratio}} = \arg\max_k \frac{\lambda_k}{\lambda_{k+1}}, \tag{82}$$

which marks the largest relative decrease in variance between consecutive components. As shown in Table 11, this diagnostic agrees with the expected active dimension in all eleven settings, including

Table 10: **Sensitivity of active dimension recovery to PVE threshold.** Each entry reports the number of training seeds recovering the expected DoF out of three. The first five rows use simulated trajectories, while the final two use real videos.

| System | 90% | 95% | 96% | 97% | 98% | 99% |
|---|---|---|---|---|---|---|
| Mass Spring | 3/3 | 3/3 | 3/3 | 3/3 | 3/3 | 0/3 |
| Single Pend. | 3/3 | 3/3 | 3/3 | 3/3 | 1/3 | 0/3 |
| Double Pend. | 3/3 | 3/3 | 3/3 | 3/3 | 0/3 | 0/3 |
| Two Body | 3/3 | 3/3 | 1/3 | 0/3 | 0/3 | 0/3 |
| Three Body | 3/3 | 3/3 | 3/3 | 3/3 | 3/3 | 3/3 |
| Single Pend. Video | 3/3 | 3/3 | 3/3 | 3/3 | 3/3 | 0/3 |
| Double Pend. Video | 3/3 | 3/3 | 1/3 | 0/3 | 0/3 | 0/3 |

Table 11: **Eigenvalue-ratio diagnostic of the recovered active dimension.** We report ratios $\lambda_k/\lambda_{k+1}$ between consecutive PCA eigenvalues. Highlights the largest ratio, whose index agrees with the expected active dimension derived from PVE threshold as well as the analytical DoF.

| System | Mode | Source | DoF | $k=1$ | $k=2$ | $k=3$ |
|---|---|---|---|---|---|---|
| Mass Spring | Traj. | Sim | 1 | 205.9 | 2.7 | 1.3 |
| Single Pend. | Traj. | Sim | 1 | 123.0 | 1.4 | 1.5 |
| Double Pend. | Traj. | Sim | 2 | 4.3 | 20.4 | 1.1 |
| Two Body | Traj. | Sim | 1 | 72.6 | 1.3 | 1.4 |
| Three Body | Traj. | Sim | 2 | 4.7 | 72.5 | 1.7 |
| Damped Pend. | Traj. | Sim | 1 | 74.8 | 1.6 | 1.1 |
| Forced Duffing | Traj. | Sim | 1 | 94.6 | 1.3 | 1.7 |
| Single Pend. | Traj. | Real | 1 | 50.4 | 2.7 | 1.8 |
| Double Pend. | Traj. | Real | 2 | 1.7 | 18.1 | 2.9 |
| Single Pend. | Video | Real | 1 | 154.2 | 1.4 | 2.9 |
| Double Pend. | Video | Real | 2 | 2.9 | 28.0 | 1.0 |

non-conservative systems and real trajectory and video observations. The agreement corroborates the dimensions obtained with the 95% criterion without requiring a cumulative variance cutoff.

**Inference-Time Pruning of Inactive Components.** We test whether the identified active subspace retains the information needed for accurate trajectory generation. At inference, we zero the inactive PCA components and evaluate trajectory MSE across the five conservative systems. As shown in Table 12, the pruned model maintains accuracy close to full SPS-GAN and outperforms the best supervised baseline on four of the five systems. These results provide functional support for the recovered dimension, as removing the components classified as inactive has only a limited effect on generated motion.

### C.2 Ablation Study on Phase Space Dynamics

Inductive biases have long shown to help neural networks with generalizability and robustness to out-of-distribution shifts [52, 53, 54, 55, 10, 12] on a wide range of machine learning applications. To better understand the role that the port Hamiltonian formulation plays in SPS-GAN, We replacing the phase space dynamics with either GRU [48] or SympNet [47]. The two ablation architectures represents two levels of ablation, where replacing pHNN with SympNet still preserves the symplectic nature of the model, while replacing pHNN with GRU reduces the model down to a black box.

While replacing pHNN with GRU and SympNet are straight forward from a network perspective, careful consideration must be taken to ensure that the change in performance originates solely from the change in dynamics modeling. The key challenge is how the phase space dynamics interfaces with the cyclic coordinate loss. Recall that under the conservative generalization of $\hat{D} = \hat{F} = 0$, the cyclic coordinate loss for conservative systems [17] is defined as

$$\mathcal{L}_{\text{cyclic}} = \lambda_{\text{cyclic}} \sum_{i=1}^{d_{\text{lat}}} |\dot{p}_i| \,. \tag{83}$$

The cyclic coordinate loss can be naively implemented using pHNN as it directly outputs derivatives of momentum. Contrast this to SympNet or GRU that provides state predictions $(\mathbf{q}_{t+1}, \mathbf{p}_{t+1})$ directly from current state $(\mathbf{q}_t, \mathbf{p}_t)$. Therefore, implementing the cyclic coordinate loss with both SympNet and GRU requires estimating the time derivatives. Therefore, we apply the cyclic coordinate loss as

$$\tilde{\mathcal{L}}_{\text{cyclic}} = \lambda_{\text{cyclic}} \sum_{i=1}^{d_{\text{lat}}} \sum_{t=0}^{T-1} \left| \frac{p_{i,t+1} - p_{i,t}}{\Delta t} \right| . \tag{84}$$

Table 12: **Trajectory accuracy after pruning inactive PCA components.** We report MSE for full SPS-GAN, its inference-time pruned variant, and the best supervised baseline for each system.

| Network | Mass Spring | Single Pend. | Double Pend. | Two Body | Three Body |
|---|---|---|---|---|---|
| SPS-GAN | $6.16 \times 10^{-4}$ | $4.29 \times 10^{-4}$ | $3.25 \times 10^{-2}$ | $3.07 \times 10^{-3}$ | $1.26 \times 10^{-3}$ |
| SPS-GAN Pruned | $5.79 \times 10^{-4}$ | $4.25 \times 10^{-4}$ | $3.54 \times 10^{-2}$ | $3.27 \times 10^{-3}$ | $1.33 \times 10^{-3}$ |
| Best Baseline | $1.13 \times 10^{-4}$ | $6.27 \times 10^{-4}$ | $4.66 \times 10^{-2}$ | $3.61 \times 10^{-3}$ | $1.37 \times 10^{-3}$ |

Table 13: **Sensitivity to cyclic-coordinate loss weight across conservative systems.** We report recovered degrees of freedom and trajectory MSE. All tested weights at or above $\lambda_{\text{cyclic}} = 0.03$ recover the expected active dimension for all five systems.

| | $\lambda_{\text{cyclic}}$ | Mass Spring | Single Pend. | Double Pend. | Two Body | Three Body |
|---|---|---|---|---|---|---|
| DoF | 0.10 | 1 | 1 | 2 | 1 | 2 |
| | 0.05 | 1 | 1 | 2 | 1 | 2 |
| | 0.03 | 1 | 1 | 2 | 1 | 2 |
| | 0.01 | 2 | 2 | 3 | 2 | 2 |
| | 0.00 | 4 | 5 | 9 | 3 | 4 |
| MSE | 0.10 | $6.0 \times 10^{-4}$ | $4.6 \times 10^{-4}$ | $3.7 \times 10^{-2}$ | $3.3 \times 10^{-3}$ | $1.4 \times 10^{-3}$ |
| | 0.05 | $6.4 \times 10^{-4}$ | $4.8 \times 10^{-4}$ | $3.2 \times 10^{-2}$ | $2.9 \times 10^{-3}$ | $1.1 \times 10^{-3}$ |
| | 0.03 | $6.2 \times 10^{-4}$ | $4.3 \times 10^{-4}$ | $3.3 \times 10^{-2}$ | $3.1 \times 10^{-3}$ | $1.3 \times 10^{-3}$ |
| | 0.01 | $8.6 \times 10^{-4}$ | $4.7 \times 10^{-4}$ | $4.0 \times 10^{-2}$ | $3.8 \times 10^{-3}$ | $1.7 \times 10^{-3}$ |
| | 0.00 | $8.9 \times 10^{-4}$ | $6.9 \times 10^{-4}$ | $6.9 \times 10^{-2}$ | $4.4 \times 10^{-2}$ | $4.4 \times 10^{-3}$ |

As shown in Table 6, SPS-GAN with GRU or SympNet phase space dynamcis suffers from both degraded prediction accuracy as well as collapse of interpretability.

### C.3 Ablation Study on Cyclic Coordinate Loss

**Sensitivity to Cyclic-Coordinate Loss.** The dimension of the phase space for SPS-GAN is primiarily regularized through the cyclic coordinate loss, defined by

$$\mathcal{L}_{\text{cyclic}} = \lambda_{\text{cyclic}} \sum_{i=1}^{d_{\text{lat}}} \left| \dot{p}_i + \left[ \mathbf{D}_\phi(\xi) \nabla_{\mathbf{p}} \mathcal{H}_\theta(\mathbf{z}, \xi) - \mathbf{F}_\psi(\mathbf{u}, \xi) \right]_i \right|. \tag{85}$$

The role of the cyclic coordinate loss is there for to identify the minimally dimension phase space and non-conservative components that is able to accurately model the system dynamics. The choice of $\lambda_{\text{cyclic}}$ is robust across systems and modalities, as all experiment reported used $\lambda_{\text{cyclic}} = 0.03$. We ablate the cyclic coordinate loss on all five conservative systems by progressively reducing its weight from 0.10 until the loss is removed in Table 13. All tested weights at or above 0.03 recover the expected active dimension with similar MSE, while lower weights can overestimate the active dimension and degrade trajectory accuracy.

**Sensitivity to Non-Conservative Regularization.** We jointly vary the dissipation and forcing regularization weights on the damped pendulum, setting $\lambda_D = \lambda_F$ (all other experiments in the paper uses $\lambda_F = 0.01$). As shown in Table 14, every tested nonzero weight recovers the expected one degree of freedom with similar trajectory MSE across two orders of magnitude. Removing both penalties increases the recovered dimension to three and substantially increases MSE. The shared value of 0.01 used for both weights throughout the dissipative and forced experiments lies within this stable range.

### C.4 Ablation Study on Phase Space Map

The phase space map projects the random motion samples $\varepsilon_{\text{m}}$ onto the phase space to generate the initial conditions for the rollout. This allows the port Hamiltonian formulation to originate from coordinates that are well condition (and in the ideal setting on the motion manifold of the system). We verify this claim by ablating the phase space map, where SPS-GAN is no longer able to identify a minimal phase space matching system degrees of freedom (see Table 6).

Table 14: **Sensitivity to non-conservative regularization weights.** We jointly vary $\lambda_D = \lambda_F$ on the damped pendulum and report recovered degrees of freedom and trajectory MSE.

| $\lambda_D = \lambda_F$ | DoF | MSE |
|---|---|---|
| 0.1 | 1 | $4.7 \times 10^{-4}$ |
| 0.01 | 1 | $4.6 \times 10^{-4}$ |
| 0.001 | 1 | $4.9 \times 10^{-4}$ |
| 0 | 3 | $2.0 \times 10^{-3}$ |

Table 15: **Robustness to observation noise.** We report the cumulative PVE of the first two components ($\text{CPVE}_2$) for double-pendulum videos across three training seeds. An SNR of $\infty$ denotes no added noise.

| SNR (dB) | Seed 72 | Seed 172 | Seed 272 |
|---|---|---|---|
| $\infty$ | 95.10% | 95.21% | 96.33% |
| 30 | 95.47% | 95.77% | 95.97% |
| 20 | 95.20% | 96.27% | 95.60% |
| 10 | 95.33% | 95.13% | 95.45% |
| 0 | 86.10% | 82.07% | 81.41% |

### C.5 Robustness to Observation Noise

We evaluate the robustness of active dimension recovery by adding per-pixel i.i.d. Gaussian noise to normalized double-pendulum videos. We retrain SPS-GAN for 20,000 steps at each signal-to-noise ratio (SNR) using three training seeds (72, 172, 272). As shown in Table 15, all three seeds recover the expected two degrees of freedom at the tested SNRs down to 10 dB. At 0 dB, where signal and noise have equal power, the cumulative PVE of the first two components falls below 95% for every seed, causing the recovered dimension to exceed the expected two degrees of freedom. These results demonstrate robustness over the tested 10–30 dB range and identify a breakdown regime at 0 dB.

### C.6 Training Stability and Initial-Condition Coverage

**Training stability.** We evaluate whether the recovered active dimension remains stable as training progresses on real double-pendulum videos in Table 16. The 95% cumulative PVE criterion recovers the expected two degrees of freedom at 102 of 106 checkpoints from step 9,300 onward, with the four exceptions confined to steps 9,400–10,400. At the checkpoints reported below, the cumulative PVE of the first two components remains above 95% while FVD decreases, indicating that video quality continues to improve while the recovered dimension remains unchanged. The active PCA directions also remain aligned across steps 10,000, 15,000, and 20,000, with cosine similarities of 0.97–0.99.

**Initial-condition coverage.** We assess whether generated initial conditions cover the underlying distribution across the five conservative systems in Table 17. We compare generated and reference initial conditions using forward KL divergence and sliced Wasserstein distance (SWD), with real-to-real comparisons providing a reference floor for each metric. Both distances remain within a small factor of their respective floors across all five systems. These results support broad coverage of the initial-condition distributions and provide evidence against mode collapse in the evaluated settings.

## D Experiment Statistical Variance

In this section we report the statistical variance for SPS-GAN and baselines methods for experiments reported in Table 1 and Table 3 in Table 18 and Table 19 respectively. We calculate the standard deviation of the results reported based on three training seeds (72, 172, 272), using the same original training data. We calculate the mean and standard deviation of the MSE such that

$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{3}\mu_i, \qquad \sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{3}(\mu_i - \bar{\mu})^2}. \tag{86}$$

## E Broader Impact

SPS-GAN may support scientific discovery and engineering by learning compact, interpretable representations of observed dynamics without prescribed state variables or complete analytical

Table 16: **Training stability on real double-pendulum videos.** We report cumulative PVE of the first two components and FVD at selected training checkpoints.

| Training step | 10k | 12k | 14k | 16k | 18k | 20k |
|---|---|---|---|---|---|---|
| $\text{CPVE}_2$ (%) | 95.39 | 95.17 | 95.75 | 96.15 | 95.19 | 95.10 |
| FVD | 194.75 | 190.90 | 152.66 | 88.41 | 78.06 | 74.69 |

Table 17: **Distributional coverage of generated initial conditions.** We report forward KL divergence and SWD between generated and reference initial conditions, alongside their real-to-real reference floors.

| Metric | Mass Spring | Single Pend. | Double Pend. | Two Body | Three Body |
|---|---|---|---|---|---|
| KL floor | 0.05 | 0.04 | 0.44 | 0.27 | 0.32 |
| KL | 0.07 | 0.06 | 0.99 | 0.41 | 0.75 |
| SWD floor | 0.073 | 0.080 | 0.085 | 0.092 | 0.109 |
| SWD | 0.090 | 0.093 | 0.121 | 0.107 | 0.117 |

models. Its generated trajectories and videos could aid hypothesis exploration and synthetic data generation where physical experiments are costly or difficult to repeat.

Physically accurate generative modeling could also be misused to make fabricated depictions of physical events more convincing. Physically plausible outputs may further be mistaken for validated predictions, since mechanical structure does not guarantee accuracy outside the observed conditions. While we do not believe SPS-GAN meaningfully changes the risk landscape relative to current generative methods, generated data should be distinguished from measurements and validated against independent observations before informing consequential scientific or engineering decisions.

Table 18: **Trajectory generation accuracy on conservative systems.** The MSE of SPS-GAN generated trajectory is compared to predicted trajectory from baselines. SPS-GAN models single systems, while cSPS-GAN jointly models all five conditionally. When generating single system dynamics, unsupervised SPS-GAN outperforms supervised baselines. When jointly modeling systems, supervised conditional cSPS-GAN performs on par with single system baselines.

| Network | Mass Spring | Single Pend. | Double Pend. |
|---|---|---|---|
| SPS-GAN | $(6.16 \pm 0.34) \times 10^{-4}$ | $\mathbf{(4.29 \pm 0.64) \times 10^{-4}}$ | $\mathbf{(3.25 \pm 0.14) \times 10^{-2}}$ |
| HNN | $(3.53 \pm 0.15) \times 10^{-4}$ | $(6.27 \pm 0.32) \times 10^{-4}$ | $(1.25 \pm 0.89) \times 10^{-1}$ |
| SymODEN | $(6.40 \pm 0.36) \times 10^{-4}$ | $(6.56 \pm 0.45) \times 10^{-4}$ | $(9.89 \pm 0.51) \times 10^{-2}$ |
| TRS-ODEN | $(6.72 \pm 0.68) \times 10^{-4}$ | $(9.57 \pm 1.87) \times 10^{-4}$ | $(6.12 \pm 0.31) \times 10^{-2}$ |
| NeuralODE | $(1.62 \pm 0.60) \times 10^{-4}$ | $(1.59 \pm 1.22) \times 10^{-3}$ | $(5.99 \pm 1.32) \times 10^{-2}$ |
| LatentODE | $\mathbf{(1.13 \pm 0.03) \times 10^{-4}}$ | $(9.27 \pm 1.06) \times 10^{-4}$ | $(4.66 \pm 0.49) \times 10^{-2}$ |
| cSPS-GAN | $(9.88 \pm 0.32) \times 10^{-4}$ | $(5.11 \pm 0.25) \times 10^{-4}$ | $(3.33 \pm 0.63) \times 10^{-2}$ |

| Network | Two Body | Three Body |
|---|---|---|
| SPS-GAN | $\mathbf{(3.07 \pm 0.15) \times 10^{-3}}$ | $\mathbf{(1.26 \pm 0.04) \times 10^{-3}}$ |
| HNN | $(3.61 \pm 0.18) \times 10^{-3}$ | $(2.81 \pm 0.08) \times 10^{-3}$ |
| SymODEN | $(6.20 \pm 0.25) \times 10^{-3}$ | $(8.08 \pm 2.85) \times 10^{-3}$ |
| TRS-ODEN | $(6.36 \pm 1.13) \times 10^{-3}$ | $(6.22 \pm 0.36) \times 10^{-3}$ |
| NeuralODE | $(5.72 \pm 0.37) \times 10^{-3}$ | $(3.67 \pm 0.57) \times 10^{-3}$ |
| LatentODE | $(1.98 \pm 0.98) \times 10^{-2}$ | $(1.37 \pm 0.10) \times 10^{-3}$ |
| cSPS-GAN | $(4.46 \pm 0.17) \times 10^{-3}$ | $(2.61 \pm 0.11) \times 10^{-3}$ |

Table 19: **Trajectory generation accuracy on non-conservative systems.** The MSE error of SPS-GAN generated trajectory is reported compared to predicted trajectory from baselines. Unsupervised SPS-GAN outperforms supervised single-system baselines.

| Network | Ideal Pend. | Damped Pend. | Forced Duffing |
|---|---|---|---|
| SPS-GAN | $\mathbf{(4.77 \pm 0.13) \times 10^{-4}}$ | $\mathbf{(4.57 \pm 0.06) \times 10^{-4}}$ | $\mathbf{(5.22 \pm 0.62) \times 10^{-4}}$ |
| pHNN | $(8.42 \pm 0.88) \times 10^{-4}$ | $(4.76 \pm 0.19) \times 10^{-4}$ | $(6.92 \pm 0.84) \times 10^{-4}$ |
| NeuralODE | $(1.59 \pm 1.22) \times 10^{-3}$ | $(6.85 \pm 0.57) \times 10^{-4}$ | $(7.35 \pm 0.59) \times 10^{-4}$ |
| LatentODE | $(9.27 \pm 1.06) \times 10^{-4}$ | $(6.69 \pm 0.27) \times 10^{-4}$ | $(1.18 \pm 0.31) \times 10^{-3}$ |
| TRS-ODEN | $(9.57 \pm 1.87) \times 10^{-4}$ | $(1.44 \pm 0.32) \times 10^{-3}$ | $(6.50 \pm 0.38) \times 10^{-4}$ |

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper claims four key contributions in the introduction, namely:

   (a) Generates physically consistent rollouts across multiple systems, parameter variations, and observation modalities (Cartesian coordinates and video);
   (b) Adopts a port-Hamiltonian formulation to model conservative, dissipative, and forced systems within a single framework;
   (c) Identifies the degrees of freedom of a system from observations alone, through a cyclic-coordinate regularizer that compresses the phase space to its minimally active dimensions;
   (d) Recovers physically meaningful canonical coordinates by decoding along the active phase space dimensions, connecting learned representations to underlying physical quantities.

   We show that

   (a) SPS-GAN achieves improved dynamical accuracy in Table 1, 3, and 4, where SPS-GAN outperforms competitive baselines;
   (b) SPS-GAN extends to non-conservative systems using simulated damped pendulum and forced duffing, as well as real world single and double pendulum, both in terms of trajectory and video, where SPS-GAN is able to faithfully recover both the degrees of freedom as well as the canonical coordinates;
   (c) SPS-GAN correctly identifies the system degrees of freedom over 11 systems in Table 5;
   (d) SPS-GAN correctly identifies the system canonical coordinates for 6 systems in Figure 3, 8a, and 8b;

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have included limitations after the conclusion of the paper, where we highlight that SPS-GAN can only accommodate systems that are Hamiltonian or port-Hamiltonian, as well as videos that exhibit approximately planar moiton.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We include detailed derivations of the system degrees of freedom and canonical coordinates in Appendix B.4.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We report the experimental setting and hyper-parameters in Appendix A. The training and inference code is open-sourced in the GitHub link below.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: Open source code is provided in the following GitHub repo: `https://github.com/yy0555/SPS-GAN`.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We include all training details in Appendix A as well as data generation procedure and governing equations. All experiments that use existing datasets are cited when first introduced.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include standard deviations for the numerical results in Appendix D. We do not include error statistics for qualitative results such as decoded canonical coordinates.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes include the compute requirement in Appendix A. Specifically, all training of SPS-GAN was performed on cluster nodes each with 1 Nvidia L40 GPU, 16 core partition of an Intel Xeon Gold 5320, and 64G of DDR4 3200MHz RDIMM.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the code of ethics and identified no deviations. We acknowledge the general harm that generative video models can do, but we do not believe that SPS-GAN poses increased risk compared to contemporary models.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Due to page limitation, we call-out that physically accuracy generative modeling may contribute to malicious use of genAI in the appendix.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We do not release data in this paper. We will include additional discussion when the github repo is made public.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing datasets are cited when first appearing in the paper. All datasets used are according to their release copy right.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: this paper does not release new assets.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not contain human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper is not subject to IRB as we do not study live subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is not included in the core contribution of the paper.